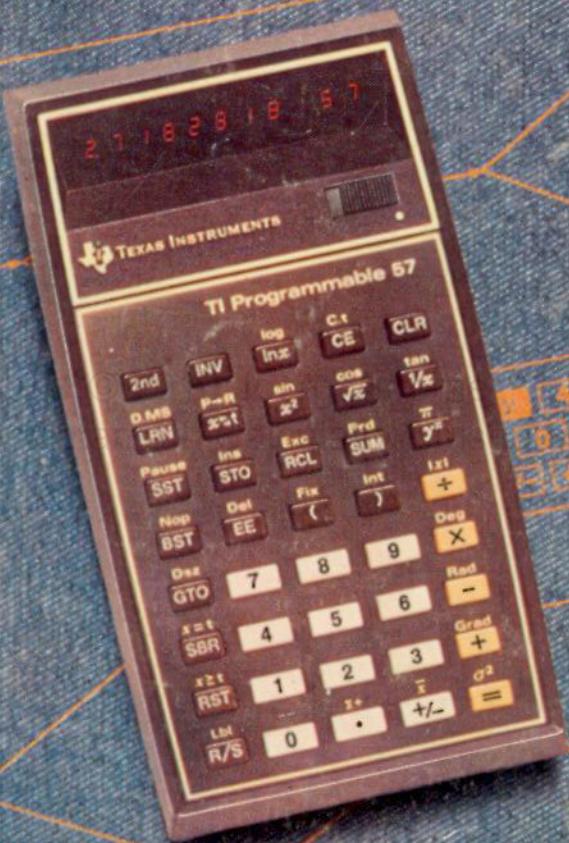


TI PROGRAMMABLE 57

DER WEG ZUM PROGRAMMIEREN



TEXAS INSTRUMENTS



TASTENINDEX

Schneller Überblick, auf welcher Seite die Beschreibung jeder Taste zu finden ist.

2nd	1-3 2-1	INV	2-1	log	2-14	C/I	2-3	CE	2-3	CLR	2-3
D.MS	2-19	F\RightarrowI	2-20	sin	2-17	cos	2-17	tan	2-17	1/x	2-12
LRN	1-2 3-2	x:t	2-22 3-8	x²	2-12	\sqrt{x}	2-12	y^x	2-13	π	2-12
Pause	1-3 3-3	Int	3-15	Exx	2-8	hd	2-8	Int	2-15	 x 	2-12
SST	3-14	STO	1-7 2-7	RCL	2-7 3-8	SUM	2-8)	2-6 4-11	$\frac{1}{x}$	2-4
Nop	3-15	Def	3-16 2-10	FIX	2-11	Int	2-15	 x 	2-12	Deg	2-16
BST	3-15	EE		(2-6)	2-6	\times	2-4	Rad	2-16
BSZ	3-6			8		9		-	2-4	Grad	2-16
GTO	3-5 3-15	7		5		6		+	2-4	Lst	2-16
x=I	3-8			2		3		Σ	2-23	o2	2-23
SBR	3-11	4		1		0		+/—	2-4	=	2-4
x=I	3-10			Z+	2-22						
RST	1-3 3-2	1		*	2-4						
Lbl	3-4										
R/S	3-3	0									

© 2010 Joerg Woerner
Datamath Calculator Museum

WICHTIG

Tragen Sie hier das Kaufdatum und die Seriennummer ein, die am Bodenteil des Rechners angebracht ist. Die Seriennummer ist durch die Worte "SERIAL NO" gekennzeichnet. Geben Sie diese Informationen bei jeder Korrespondenz an.

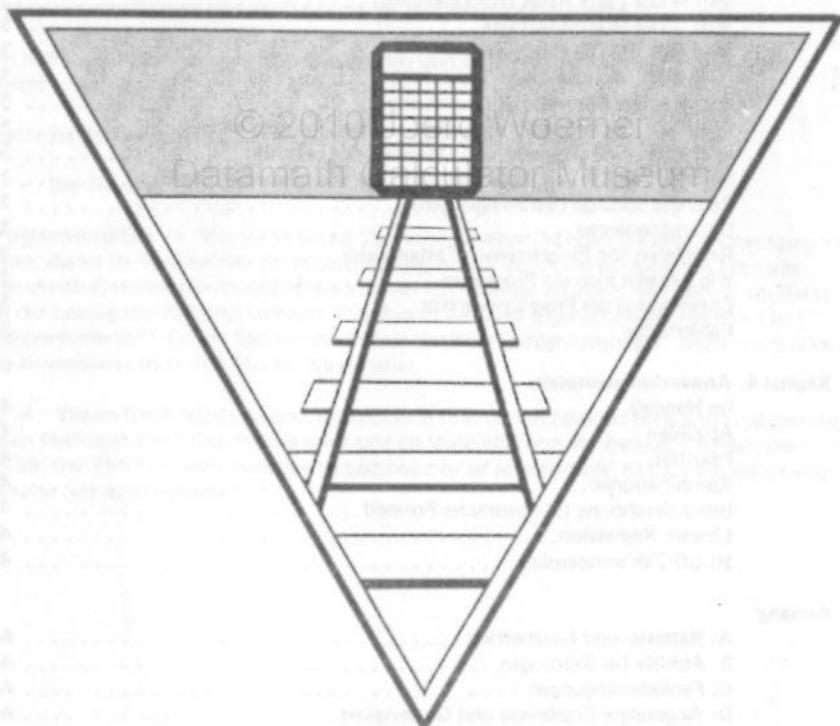
PROGRAMMIERRARER RECHNER TI-57

Modell-Nummer

Serien-Nummer

Kaufdatum

DER WEG ZUM PROGRAMMIEREN



INHALT

Tastenindex	Umschlagseite innen
Inhalt	i
Kapitel 1: Einführung	
Ein erster Blick auf den programmierbaren Rechner TI-57.	1-1
Wie funktioniert der Rechner?	1-2
Ein praktisches Beispiel	1-6
Kapitel 2: Weitere Informationen – eine Beschreibung der Tasten und Funktionen	
Abschnitt 1: Grundfunktionen und Tasten.	2-2
Abschnitt 2: Technisch-wissenschaftliche Funktionen und Tasten.	2-10
Abschnitt 3: Statistische Funktionen und Tasten.	2-22
Kapitel 3: Eine Beschreibung der Programmertasten	
LRN – Die Taste Learn.	3-2
RST – Die Taste Reset (Rückstellung).	3-2
R/S – Die Run/Stop-Taste	3-3
2nd Paus – Die Pausentaste	3-3
2nd Lbl n – Die Label-Tastenfolge	3-4
GTO n – Der Sprungbefehl zum Label.	3-5
GTO 2nd nn – Der Sprungbefehl zu einer Schrittzahl	3-5
2nd DLT – Dekrement und Überspringen bei Null	3-6
INV 2nd DLT – Dekrement und Überspringen bei Nicht-Null	3-7
Bedingte Sprünge (Verzweigungen)	3-8
Unterprogramme	3-11
Redigieren von Programmen: Tastenkodes	3-13
Wie schreibt man ein Programm	3-16
Entwicklung des Programmierstils	3-17
Fehlersuche	3-18
Kapitel 4: Anwendungsbeispiele	
Im Handel.	4-1
Abzählen	4-3
Fakultät	4-6
Kombinationen	4-9
Integralrechnung (Simpsonsche Formel)	4-11
Lineare Regression.	4-15
HI-LO-Zahlenratespiel	4-21
Anhang:	
A: Batterie- und Netzbetrieb	A-1
B: Abhilfe bei Störungen	A-4
C: Fehlerbedingungen	A-6
D: Angezeigte Ergebnisse und Genauigkeit	A-8
E: Tastenkodes	A-9
Lösungsvorschläge	A-10

EIN ERSTER BLICK AUF DEN PROGRAMMIERBAREN RECHNER TI-57

Sie beginnen einen Rechner zu benutzen, der gerade aus der Welt der Zukunftsvisionen kommt. Erst die jüngsten Fortschritte in der Solid-State-Technologie machten es möglich, dieses Bündel an Leistungsfähigkeit in Ihre Hände zu legen, das zugleich einfach in der Anwendung ist und Ihnen sicher Vergnügen bereitet.

Einfache Anwendung

Ein kurzer Blick auf den Rechner sagt Ihnen bereits, daß er mit einer ganzen Menge von Tasten und Eigenschaften ausgestattet ist. Der primäre Zweck dieses Buches ist die schrittweise Einführung in diese Tastenfunktionen und Merkmale. Typisch für alle diese Eigenschaften ist jedoch ein Leistungsmerkmal Ihres programmierbaren Rechners TI-57: die Anwendung ist einfach. Ohne Probleme können Sie viele Aufgaben erledigen, die bei den Schularbeiten sehr hilfreich oder in Ihrer beruflichen Entwicklung von großem Vorteil sein können. Und im täglichen Leben bringt das Arbeiten mit dem Rechner Freude. Sie sollen damit ein Hilfsmittel erhalten, das zusammen mit den Informationen in diesem Buch zu einem echten Problemlösungs-System und der Schlüssel für Ihre Berechnungen wird.

Wir hoffen, daß Sie Freude an der Anwendung und an der Suche nach neuen Lösungen haben.

Damit Sie Ihr Gerät kennenzulernen, zunächst eine Erklärung des Namens:

TI — Der Rechner wird von Texas Instruments INC. hergestellt.

Programmierbar — Wie Sie in kurzer Zeit sehen werden, können Sie einen Lernvorgang veranlassen, damit Ihr Rechner bei der Problemlösung hilft — durch Berechnung von Formeln oder durch Erstellen von Modellen. Sie können Rechenwege angeben, nach denen das Gerät bei der Lösung der Aufgaben vorgeht. Der Begriff für diese Eigenschaft des Rechners ist „programmierbar“. Lassen Sie sich aber durch das Wort „programmierbar“ nicht erschrecken. Die Anwendung Ihres TI-57 ist ein Kinderspiel.

57 — Dieses Gerät ist die neueste Entwicklung in einer wachsenden Serie von programmierbaren Rechnern von Texas Instruments, und als Vorbild diente ein anderer Rechner, der SR-56. Der TI-57 ist noch einfacher zu bedienen; er ist so aufgebaut, daß Sie ihn sofort und mühelos benutzen können.

WIE FUNKTIONIERT DER RECHNER?

HOCHSICHTLICHE KOMPUTER
VON RECHNER VERFAßT

Auf einfache Weise! Wenn Sie Anfänger sind, nehmen Sie den Rechner heraus und prüfen Sie das Gewicht dieses kleinen Gerätes. Es ist sehr leicht, weil die wichtigsten Vorgänge im Rechner in einem sogenannten seriell orientierten arithmetischen Prozessor (SOAP-Chip) verarbeitet werden.

Schalten Sie jetzt den Rechner ein. Eine Null in der Anzeige ist der Hinweis, daß Sie mit der Arbeit beginnen können. (Wenn die Null nicht erscheint oder wenn die Anzeige in anderer Form unlogisch ist, muß die Batterie geladen werden. Schalten Sie in diesem Fall den Rechner wieder aus, stecken Sie das Ladegerät ein und warten Sie ein paar Minuten; dann können Sie Ihre Berechnungen aufnehmen.)

Beginnen wir mit einem Beispiel. Stellen Sie sich folgende Situation vor:

Sie haben von einer Radiostation soeben einen recht eigenartigen Preis gewonnen. Sie erhalten 2 Pfennig! Dieser Betrag wird drei Wochen lang jeden Tag verdoppelt (21 Tage). Oder – Sie können sich sofort 5000 DM in bar auszahlen lassen. Für Ihre Entscheidung haben Sie zwei Minuten Zeit – der Ansager wartet. (Tatsächlich handelt es sich um eine alte klassische Aufgabe. Sehen Sie, welche Hilfe Ihr Rechner bietet.)

Grundsätzlich können Sie den Rechner veranlassen, die aufgelaufenen Beträge anzuzeigen. Hier ist die Lösung:

In der oberen linken Ecke der Tastatur, in der zweiten Reihe von oben, ist die Taste **LRN**, ein reines Juwel.



Es handelt sich um die Learn-Taste, und wenn sie einmal gedrückt wird, beginnt der Rechner alle nächsten Eingaben zu „lernen und sich daran zu erinnern“. Wenn Sie **LRN** erneut drücken, weisen Sie damit den Rechner an, daß der Lernvorgang für den Augenblick abgeschlossen ist. Der Rechner beendet an dieser Stelle die Aufnahme der Informationen, „erinnert“ sich aber an alle Schritte, die Sie bisher angewiesen haben. (Die Taste **LRN** wirkt wie ein EIN/AUS-Schalter für die Lernvorgänge Ihres Rechners.)

Drücken Sie also **LRN**. (Die Anzeige verändert sich und weist **00 00** aus.) Wenn Sie sich jetzt beim Preisausschreiben für die 2-Pfennig-Alternative entscheiden, verdoppeln Sie 21 Tage lang den Betrag und Sie wollen sicher beobachten, wie er anwächst. Hierzu drücken Sie die Tasten **X 2 =**.

WIE FUNKTIONIERT DER RECHNER ?

TIFFENHOLZ
THERMOPOLAR

Nun eine weitere wertvolle Funktion – direkt über der Taste **SST** sehen Sie das Wort „Pause“ auf der Rechneroberfläche. Ist ein Wort oder eine Funktion über einer Taste auf der Rechneroberfläche und nicht auf der Taste selbst aufgedruckt, spricht man von einer „Zweitfunktion“. © 2010 Joerg Woerner

Mit Zweitfunktionen kann der Rechner mit maximaler Leistungsfähigkeit ausgestattet werden, ohne ihn mit Tasten zu überladen. Um eine Zweitfunktion zu verwenden, drücken Sie einfach die Taste **2nd** in der oberen linken Ecke und dann die Taste direkt unter der Zweitfunktion. In diesem Buch werden Tasten mit schwarzem Grund verwendet, um auf Zweitfunktionen hinzuweisen. Also gilt:

TASTE: **2nd Pause** (Tatsächlich drücken Sie **2nd** und die Taste **SST**.)

Die Taste **Pause** weist den Rechner an, einen Moment zu unterbrechen, um die Beobachtung der Rechenvorgänge zu ermöglichen.

Ein weiterer Punkt – Sie wollen den Geldbetrag wiederholt verdoppeln. Um Ihr Gerät entsprechend anzusehen, suchen Sie die Taste **RST** (reset), nahe der unteren linken Ecke. Die Taste veranlaßt die Rückstellung an den Anfang und die Wiederholung der Berechnung.

Drücken Sie: **RST**

Jetzt haben Sie den Rechner mit allen nötigen Informationen versorgt. Drücken Sie also **LRN** und dann **RST** erneut, um sicher zu sein, daß alles auf den Anfang zurückgestellt ist.

Um nun das Anwachsen des 2 Pfennig-Betrages zu beobachten, geben Sie zunächst mit den Tasten **• 0 2** die 2 Pfennige ein. (In der Anzeige muß 0.02 ausgewiesen sein.)

Jetzt ist der Rechner bereit, den Betrag zu erhöhen – jeder Anzeigewert ist der doppelte Betrag des vorhergehenden. Sie beginnen die Berechnung und zählen die Tage mit. Am ersten Tag haben Sie 0.04, etc.. Wenn Sie 21 Tage erreichen, ist der Wert in der Anzeige der Betrag, den Sie gewinnen.

Der Rechner wird einfach mit der RUN/STOP-Taste – **R/S** – gestartet. Sie befindet sich auf der Tastatur unten links. Achten Sie auf die Anzeige, zählen Sie 21 Tage aus, dann drücken Sie die Taste **R/S** erneut und halten Sie für einen Moment fest.

Am Tag 21 haben Sie 41943.04 DM gewonnen. Sagen Sie dem Sprecher, daß Sie sich für die 2 Pfennig-Alternative entschieden haben.

WIE FUNKTIONIERT DER RECHNER ?

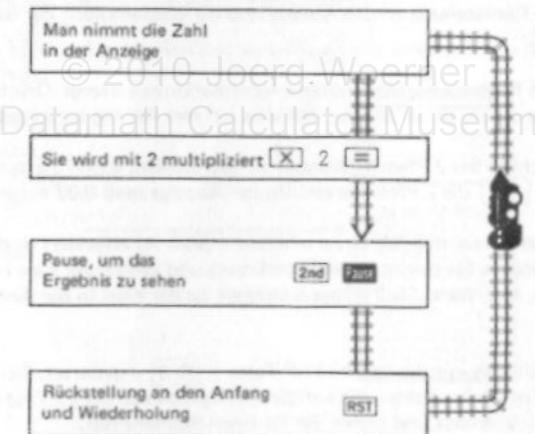
Herzlichen Glückwunsch!

Mit diesem kleinen Beispiel haben Sie bereits viele der wesentlichen Punkte erfaßt, die die Anwendung Ihres programmierbaren Rechners problemlos und zu einem Vergnügen machen.

- Mit der Taste **[LRN]** können Sie den Rechner unterweisen (Sie können bis zu 50 Schritte eingeben und er kann eine Menge „lernen“ – im allgemeinen entspricht jeder dieser Schritte einem Tastendruck, einige Schritte können zwei oder mehr Tastenbe-tätigungen umfassen.)
- Die Taste **[RST]** stellt alles auf den Anfang zurück.
- Mit der Taste **[R/S]** können Sie die Vorgänge steuern – und beliebig unterbrechen und wieder starten.

Die Rechenwege

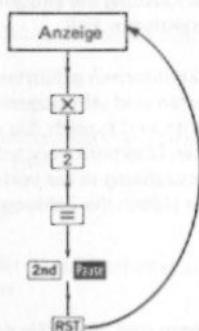
Eine Möglichkeit, sich die Funktionsweise des Rechners vor Augen zu halten ist, wenn man sich eine kleine Eisenbahn vorstellt – wo man Gleise anlegen kann, denen der Wagen folgen muß:



Der „Zug“ auf diesem Gleis hat einen besonderen Namen – der „Zeiger“ (oder, noch technischer ausgedrückt, der Programmzeiger). Jedesmal, wenn der Zug an einer Operation vorbei-kommt, befolgt der Rechner den Befehl oder führt die angegebene Operation durch. Effektiv drückt der Rechner an Ihrer Stelle die Tasten – genau in der von Ihnen bestimmten Reihen-folge. Wenn Sie die Taste **[RST]** drücken (oder wenn der Zug die Taste **[RST]** als Teil eines Programmes passiert), kehrt er sofort an den Anfang zurück – zum ersten Schritt (Schritt Nummer 00) und beginnt von neuem. Um den Zug in Bewegung zu setzen oder um ihn anzu-halten, wenn er fährt, verwenden Sie einfach die Taste **[R/S]** (Run/Stop).

WIE FUNKTIONIERT DER RECHNER ?

In dem zuvor behandelten Beispiel drückte der Rechner einfach seine eigenen Tasten und verdoppelte den Anzeigewert, wenn Sie die Taste [R/S] verwenden:



Dieser Vorgang hätte sich längere Zeit wiederholt, wenn Sie nicht nach 21 Abläufen mit der [R/S]-Taste unterbrochen hätten. (Später werden Sie sehen, daß die wiederholten Ablaufsituationen beim Rechner und auch bei größeren Computern „Schleifen“ genannt werden.)

An dieser Stelle sind Sie bereits mit vielen Merkmalen vertraut, die Ihren programmierbaren Rechner zu einer Besonderheit machen. Tatsächlich ist hier bereits genug erklärt, daß Sie jetzt alle Arten von Programmen selbst überprüfen können. Selbstverständlich werden später in diesem Buch noch alle Einzelheiten des Rechners mit weiteren Informationen über Problemlösungen und Programmabläufe eingehend behandelt. Aber scheuen Sie sich nicht, den Rechner selbstständig auszutesten. Keine Tastenkombination kann das Gerät beschädigen. Der Rechner kann auf einfache Weise programmiert werden und jede Folge von Tastenbefehlen (bis zu 50 Schritte) durchlaufen, die Sie auch manuell über die Tastatur verarbeiten können.

Zuerst drücken Sie [LRN].

- Geben Sie Ihre Tastenbefehle ein.
([RST] muß der letzte Schritt sein, wenn die Berechnung wiederholt werden soll. Falls Sie nach einem Ablauf eine Unterbrechung wünschen, muß [R/S] Ihr letzter Schritt sein. Auch wenn Sie im Verlauf eines Programms eine Zahl eingeben oder ein Zwischenergebnis sehen wollen, müssen Sie an dieser Stelle einen [R/S]-Befehl eingeben.)
- Drücken Sie [LRN] erneut.
- Drücken Sie [RST].

Jetzt haben Sie den Rechner programmiert. Um das Programm durchzuführen, gehen Sie wie folgt vor:

- Geben Sie die Zahl(en) ein, mit der (denen) Sie rechnen wollen.
- Drücken Sie [R/S].

EIN PRAKTISCHES BEISPIEL

THEMENKURS EIN
3 RECHNER ABO

Viele Arten, wie Sie Ihren programmierbaren Rechner nutzen können, werden an einer Vielzahl von Anwendungsbeispielen für Schule, Heim und Freizeit erläutert. An jedem Beispiel können Sie ermessen, mit welcher Mühe losigkeit und Leistung die alltäglichen mathematischen Probleme gelöst werden. Zunächst ein recht praktischer Fall.

In dieser Situation sei angenommen, daß auf einem kredititorisch geführten Konto eine ziemlich hohe Rechnung (500 Dollar) offen ist. Kreditkarten sind sehr bequem, aber auch die Zinsbelastung ist entsprechend hoch – 1,5% pro Monat sind typisch. Sie wollen verschiedene Zahlungsweisen gegeneinander abwägen und sich einen Überblick verschaffen, welche Alternative die beste ist. Natürlich wäre eine sofortige Rückzahlung in bar vorzuziehen, aber im Augenblick sind Sie dazu nicht in der Lage. Wenn Sie jedoch die Zahlungen auf einen längeren Zeitraum ausdehnen, werden Sie belastet.

Ihr Rechner kann die Alternativen schnell ermitteln.

Die Frage ist jetzt, wie hoch die monatlichen Zahlungen sind, wenn Sie den Betrag in 3, 6, 9 oder 12 Monaten begleichen, und wieviel Zinsen Sie in jedem Einzelfall entrichten müssen?

So müssen Sie vorgehen:

Zuerst schalten Sie den Rechner kurz aus und wieder ein, um ihn vollständig zu löschen, und dann drücken Sie die folgenden Tasten. (Denken Sie daran, daß die Erklärung der Einzelheiten erst später folgt. Für den Augenblick genügt es zu beobachten, was der Rechner Ihnen abnehmen kann. Drücken Sie jede Taste und passen Sie dabei gut auf.)

LRN
RCL 1 [X]
(RCL 2 [+]
([- ()] + RCL 2 [])
[Y] RCL 3 [+/-] () () =
R/S
[X] RCL 3 [=]
- RCL 1 [=]

R/S (An dieser Stelle ist 27 00 in der Anzeige. Wenn nicht, schalten Sie den Rechner aus (OFF) und ein (ON), gehen zurück und tasten die Folgen erneut sorgfältig ein.)

Jetzt drücken Sie:

LRN

EIN PRAKTISCHES BEISPIEL

Jetzt müssen Sie nur noch die Werte für Ihre Berechnungen in den drei verwendeten Speichern speichern. Alles übrige können Sie dem Rechner überlassen.

Geben Sie den geschuldeten Betrag ein und speichern Sie ihn wie folgt:

500 **STO** 1

Geben Sie den monatlichen Zinssatz ein ($i = 1.5\% = .015$) und speichern Sie ihn im Speicher zwei.

.015 **STO** 2

Geben Sie die erste Alternative ein, die Sie prüfen wollen – gleiche Zahlungen für $n = 3$ Monate:

3 **STO** 3

Von hier an ist es nur noch ein Kinderspiel!

Sie drücken zuerst **2nd** **Fix** 2, sodaß in der Anzeige bereits DM und Pfennig abzulesen sind.

Dann drücken Sie **R/S**.

Das Ergebnis: Wenn Sie den Betrag in drei Monaten zurückzahlen, ist die Monatsrate 171.69.

Drücken Sie **R/S** erneut, um auch die Zinsbelastung zu erfahren – in diesem Fall 15.07.

Prüfen Sie auch die anderen Möglichkeiten:

Eingabe 6 Monate – Tastenfolge 6 **STO** 3

Dann drücken Sie **R/S**

Ihre Zahlung: 87.76

R/S erneut für die Zinsbelastung: 26.58

Eingabe 9 Monate – Tastenfolge 9 **STO** 3

Dann **R/S**

Ihre Zahlung: 59.80

R/S erneut für die Zinsbelastung: 38.24

Eingabe 12 Monate – Tastenfolge 12 **STO** 3

Dann **R/S**

Ihre Zahlung: 45.84

R/S erneut für die Zinsbelastung: 50.08

EIN PRAKTISCHES BEISPIEL

Sie können jetzt fortfahren und dieses Zahlungsprogramm sehr sorgfältig ausarbeiten, um die Höhe der Zinsbelastung für jede Alternative genau zu überprüfen.

In diesem Beispiel haben Sie den Rechner für die Verarbeitung einer sehr wichtigen und vielseitigen finanz-mathematischen Formel verwendet. Die Formel für die Berechnung der (monatlich gleichen) Zahlungen für ein Konto dieses Typs lautet:

$$\text{Zahlung} = \text{PV} \times \left(\frac{i}{1 - (1+i)^{-n}} \right)$$

wobei:
i = monatlicher Zinssatz (1.5% oder .015)
n = Anzahl der Monate
PV = der Barwert des geschuldeten Betrags (500)

Die gesamte Zinsbelastung wird wie folgt berechnet:

$$\text{Zinsbelastung} = (n \times \text{Zahlung}) - \text{PV}$$

Bei der Berechnung der obigen Aufgabe wurde der Barwert der Schuld (PV) im Speicher 1 gespeichert. Der monatliche Zinssatz (i%) wurde in Speicher 2 eingebracht und die Anzahl der Monate (n) in Speicher 3.

Von nun an können Sie jede beliebige Monatsanzahl (für jeden beliebigen Barwert oder für einen beliebigen Zinssatz) überprüfen. Sie brauchen hierzu nur die Werte in den Speichern zu ändern. Mit einem Tastendruck übernimmt Ihr Rechner die gesamte Arbeit.

EIN PRAKTISCHES BEISPIEL

Zusammenfassung

An diesem Punkt haben Sie bereits eine Menge erfahren. Sie konnten den Rechner in zwei Programmsituationen beobachten — jede unter dem Aspekt der Zweckmäßigkeit und des Vergnügens. Und Sie haben hoffentlich auch gesehen, wie einfach der Rechner in der Anwendung sowie in der Programmierung ist. Ehe die Erläuterungen fortgesetzt werden, noch eine kurze Zusammenfassung über die wesentlichen Punkte:

Ihr programmierbarer Rechner ist vor allem vielseitig und leistungsfähig — alle Arten von manuellen Berechnungen können Sie jederzeit direkt über die Tastatur durchführen.

Zweitens ist Ihr Rechner programmierbar — und das bedeutet, er kann Ihre Tastenbefehle erhalten, dann an den Anfang zurückgehen und an Ihrer Stelle seine eigenen Tasten drücken. Sie haben erfahren, wie problemlos es ist, den Rechner zu programmieren, damit er nach Ihren Anweisungen vorgeht.

- Sie drücken **LRN**.
- Dann geben Sie Ihre Tastenbefehle ein.
- **LRN** wird erneut gedrückt.
- Drücken Sie **RST** (diesen Befehl dürfen Sie nicht vergessen — das Programm muß an der richtigen Stelle beginnen, und mit **RST** beginnt es beim ersten Schritt).
- Jetzt geben Sie nur noch die Zahlen ein, mit denen Sie rechnen wollen, und drücken **R/S**.

Datamath Calculator Museum

Diese neue Dimension der Programmierbarkeit bringt Ihnen einige wesentliche Vorteile. Schnellere und genauere Berechnungen — speziell in ständig wiederkehrenden Situationen — sind jetzt möglich. Es ist einfacher geworden, den Rechner als Teil von Entscheidungsprozessen einzusetzen, mathematische Beziehungen herauszufinden und ihn in einer Vielzahl alltäglicher und beruflicher Situationen anzuwenden.

Drittens, und das ist wohl der wichtigste Aspekt, soll die Arbeit mit dem Rechner Freude machen! Scheuen Sie sich nicht vor Experimenten.

EINE BESCHREIBUNG DER TASTEN UND FUNKTIONEN

Ihr Rechner ist ein leistungsfähiges Problemlösungs-System, bei dem auch auf die einfache Anwendung besonderer Wert gelegt wurde. Den Ablauf einiger simpler Programme haben Sie bereits gesehen, und in späteren Kapiteln folgt eine genaue Erläuterung der Programmierungseigenschaften. Aber neben der Programmierbarkeit ist dieses Gerät ein leistungsfähiger technisch-wissenschaftlicher Rechner. Berechnungen können jederzeit direkt über die Tastatur durchgeführt werden. In diesem Abschnitt werden die Tasten beschrieben, die unmittelbar zur Problemlösung dienen – Tasten, die Sie sofort benutzen können, um Ergebnisse zu erhalten. Weitere Abschnitte enthalten dann Erläuterungen, wie diese grundlegenden Tasten zusammen mit den anderen zum Erstellen von Programmen verwendet werden.

Die Tastatur Ihres Rechners ist logisch und unkompliziert aufgebaut.

Sie können einfache Aufgaben wie Überprüfung des Scheckhefts oder Aufaddierung der eingekauften Waren ebenso wie komplexe technische Probleme durchführen. Die Zweckmäßigkeit jeder Maschine oder jedes Werkzeugs hängt jedoch immer vom Anwender ab. Sie werden sicher wünschen, mit allen Eigenschaften vertraut zu werden – alles, was Ihnen der Rechner abnehmen kann (und was nicht).

Damit Sie Ihren Rechner voll nutzen können, nehmen Sie sich die paar Minuten Zeit, die nötig sind, um jede Taste und deren Funktion kennenzulernen.

Als Erleichterung für Sie ist diese Beschreibung in drei Hauptteile gegliedert:

Grundfunktionen und Tasten

Technisch-wissenschaftliche Funktionen und Tasten

Statistische Funktionen und Tasten

Diejenigen, die bereits mit einem dieser fortschrittlichen technisch-wissenschaftlichen Rechner vertraut sind, wollen vielleicht diesen Abschnitt überspringen und sich im Kapitel 3 sofort mit der Programmierung beschäftigen. Wenn Sie aber einige Tasten noch nicht kennen, empfiehlt sich doch die Durchsicht dieser Beschreibung. Nehmen Sie den Rechner heraus und halten Sie ihn bereit.

Zunächst einige Grundinformationen zu Ihrem Rechner, ehe die „Extras“ behandelt werden: Zu diesen Grundlagen gehört, wie die Informationen eingebracht und ausgegeben werden können – und wie der arithmetische Teil der Mathematik schnell und präzise verarbeitet werden kann.

Die Anzeige

Jedesmal, wenn Sie den Rechner einschalten, wird eine einzelne Null ausgewiesen. Das bedeutet, alles funktioniert, der Rechner ist eingeschaltet und für eine Aufgabe bereit. Durch kurzes Aus- und Einschalten des Rechners wird der Rechner vollständig gelöscht. Um die Anzeige zu testen, drücken Sie die Taste **[8]**, die Dezimalkomma-Taste **[.]**, die Vorzeichenwechseltaste **[+/-]**, und dann wiederholt die Taste **[8]**, bis jedes Segment der Anzeige aufleuchtet. Sie können bis zu 8 Ziffern gleichzeitig eintasten – dies gilt für positive und negative Zahlen. (Eingaben nach der achten Stelle werden ignoriert). Um die Genauigkeit zu sichern, werden jedoch intern alle Rechenergebnisse mit 11 Stellen berechnet und für die Anzeige auf 8 Stellen gerundet. Beachten Sie, daß zum leichteren Ablesen das Minuszeichen unmittelbar links vor einer negativen Zahl steht.

Ehe Sie die Lektüre dieser Beschreibung fortsetzen, drücken Sie die Taste **[CLR]** in der oberen rechten Ecke der Tastatur.

2nd und INV : Die Doppelfunktionstasten

Ihr Rechner verfügt über eine große Anzahl von Funktionen, mit denen alle Arten von Berechnungen mühelos und mit großer Genauigkeit ausgeführt werden können. Um den Zugriff auf diese umfangreiche Leistung zu ermöglichen, ohne dabei den Rechner mit Tasten zu überladen, haben viele Tasten mehr als eine Funktion. Die Erstfunktion ist auf der Taste selbst aufgedruckt. Man verwendet die Erstfunktion jeder Taste, wenn man einfach die Taste selbst drückt. Die Zweitfunktion ist direkt über einer Taste angegeben. Wenn Sie die Zweitfunktion einer Taste anwenden wollen, drücken Sie **[2nd]** (oben links auf der Tastatur) und dann die Taste direkt unter der gewünschten Funktion. (Zweitfunktionen werden in diesem Buch durch Tastensymbole mit schwarzem Grund gekennzeichnet, **[π]**. Die Zahl π wird zum Beispiel mit der Folge **[2nd]** **[π]** in die Anzeige gebracht.)

Auch die Umkehrfunktionstaste **[INV]** ermöglicht zusätzliche Rechnerfunktionen, ohne die Anzahl der Tasten zu erhöhen. Die Taste **[INV]** kehrt den Zweck oder die Funktion bestimmter Tasten um. **Anmerkung:** In Folgen, bei denen Sie beide Tasten, **[INV]** und **[2nd]** benötigen – ist die Reihenfolge ihrer Anwendung beliebig. Sie erhalten immer das gleiche Resultat. Drückt man **[INV]** vor einer Funktion, die keine Umkehrung besitzt, wird dieser Befehl einfach ignoriert.

ABSCHNITT 1: GRUNDFUNKTIONEN UND TASTEN

Löschen des Rechners

Es gibt mehrere Verfahren, mit denen man verschiedene Teile des Rechners bzw. das gesamte Gerät löschen kann – je nach Ihrem Bedarf, wenn Sie eine Aufgabe durchführen.

- [CE]** – Die Eingabelöschtaste **[CE]** (clear entry) löscht die zuletzt eingegebene Zahl, solange dieser Zahl noch keine Funktions- oder Operationstaste folgt. (Wenn Sie also mitten in einer Eingabe aus Versehen **5** statt **6** drücken, korrigieren Sie mit **[CE]** und geben die Zahl erneut ein.) Mit dieser Taste stellen Sie auch das Blinken der Anzeige ab, wenn Sie im Rechner eine Fehlerbedingung verursacht haben (mehr darüber später). Die **[CE]**-Taste beeinflusst dagegen keine laufenden Operationen, auch nicht den Speicherinhalt oder Rechenergebnisse.
- [CLR]** – Die Löschtaste **[CLR]** (clear) oben rechts auf dem Gerät löscht den Rechner vollständig, ausgenommen gespeicherte statistische Daten, Einstellungen des Anzeigeformats und Programmschritte.
- [2nd]** **C.t** – Mit dieser Taste kann man allein das „*t*“-Register (oder den „*t*“-Speicher) löschen, eine Eigenschaft, die besonders in der Programmierung und bei statistischen Funktionen von Vorteil ist. Detaillierte Informationen folgen später.
- [INV]** **[2nd]** **C.t** – Diese Tastenfolge löscht den gesamten Rechner mit Ausnahme der Programmschritte und der Dezimalkommaleinstellungen. (Technisch heißt das, daß die beiden ersten mathematischen Stackregister nicht gelöscht werden, aber dies führt selten zu Problemen.)

AUS-EIN (OFF-ON) – Durch kurzes Aus- und wieder Einschalten wird der gesamte Rechner gelöscht. Diese Maßnahme ist eine sichere Möglichkeit, alle Programmschritte, Speicher, Anzeige-Einstellungen, kurz, alles zu löschen.

ABSCHNITT 1:

GRUNDFUNKTIONEN UND TASTEN

0 bis 9 • +/ - – Dateneingabetasten

Der Rechner arbeitet mit Fließkomma und die Zahleneingabe erfolgt mit den Dateneingabetasten **0 – 9**, **.** und **+/-**. Bei Eingabe einer Zahl bleibt das Dezimalkomma rechts der Eingabe, bis Sie die Dezimalkommataste (**.**) drücken. Danach wird der Bruchteil der Zahl eingetastet, und das Dezimalkomma rückt entsprechend nach links. Zur Änderung des Vorzeichens einer Zahl in der Anzeige drückt man die Vorzeichenwechsel-Taste einmal. (Durch erneutes Drücken von **+/-** erhält das Vorzeichen wieder den ursprünglichen Status.)

+ - X ÷ und = – Tasten für die Grundfunktionen

Die Arithmetik wird mit 5 Operationstasten durchgeführt: **+**, **-**, **X**, **÷** und **=**. Der Rechner verfügt über ein einfaches, aber leistungsfähiges Eingabesystem – die AOS™ – Eingabemethode. Mit diesem System wird die Lösung von Problemen erheblich leichter. Sie geben die Aufgabe in der Form des schriftlichen Ansatzes ein, drücken **=** und erhalten Ihr Ergebnis. Das erstaunliche Merkmal der AOS-Methode: Gemischte Operationen werden automatisch sortiert und für die Berechnung des Resultats in der richtigen Reihenfolge verarbeitet. (Mehr Informationen über die AOS-Eingabemethode finden Sie im nächsten Abschnitt.)

© 2010 Jörg Woerner

Wenn Sie die Taste **=** drücken, werden alle unvollständigen Operationen (Vorgänge, die rechner-intern noch offen sind) abgeschlossen. Sie erhalten Ihr Ergebnis, und der Rechner wird gelöscht – und ist für eine neue Aufgabe vorbereitet.

Beispiel: Berechnen Sie $15 + 7 \times 31 - 4 = ?$

Tasten	Anzeige
15 + 7 X 31	
= 4 =	228

Anmerkung: Mit der AOS-Eingabemethode erhalten Sie ohne Schwierigkeiten das richtige Ergebnis in diesem Beispiel. Nicht alle Rechner haben diese Methode.

ABSCHNITT 1: GRUNDFUNKTIONEN UND TASTEN

Die AOS™ Eingabemethode

Die Mathematik ist eine Wissenschaft, in der eine Vielzahl von Regeln eingehalten wird. Eine dieser Regeln besagt, daß für die gleiche Operationsfolge niemals zwei verschiedene Antworten erlaubt sind. Wegen dieser Forderung – eine Lösung für eine Berechnung – wurde eine Reihe allgemein gültiger Regeln entwickelt, wenn in einer Berechnung gemischte Operationen verwendet werden. Die Aufgabe

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

hat daher nur eine richtige Lösung! (Kennen Sie das Ergebnis? Es ist 9.)

Sie können die Aufgabe direkt von links nach rechts in den Rechner eintasten, der mit dem AOS-Eingabesystem arbeitet. Und Sie erhalten das richtige Ergebnis. Der Rechner sortiert die Operationen, die Sie eingeben, wendet sie in der richtigen Reihenfolge an, und Sie können die einzelnen Rechengänge beobachten. Das AOS-Eingabesystem wirkt wie ein Organisator! Die von Ihnen angewiesenen Operationen werden geordnet und in der nachstehenden allgemein geltenden Reihenfolge ausgeführt:

- 1) Sonderfunktionen mit einer Variablen (wie x^2 , \sqrt{x} etc.) – sie wirken unmittelbar auf die angezeigte Zahl, sobald Sie die entsprechende Taste drücken. (Mehr über diese Tasten erfahren Sie an späterer Stelle in dieser Beschreibung – aber zu dieser Gruppe gehören alle Tasten für die trigonometrischen und logarithmischen Funktionen und deren Umkehrungen, sowie Quadratwurzel- und Reziprokwert-Tasten.)
- 2) Exponentialberechnungen y^x und INV (y^x (oder $\sqrt[3]{y}$) werden anschließend durchgeführt. (Weitere Informationen in einem späteren Abschnitt).
- 3) Es folgen Multiplikationen und Divisionen, in der Reihenfolge von links nach rechts, und dann
- 4) Additionen und Subtraktionen, ebenfalls von links nach rechts.

Die Gleichheitstaste $=$ schließt alle Operationen ab.

Als Sie die Grundschule noch besuchten, wurde als Gedächtnissstütze vielleicht von „Punkt-vor-Strich-Rechnung“ gesprochen, um an den letzten Teil der Hierarchie zu erinnern: Zuerst Multiplikationen und Divisionen, von links nach rechts – und dann Additionen und Subtraktionen. Ein Rechner, der mit dem AOS-Eingabesystem arbeitet, beachtet an Ihrer Stelle alle diese Regeln.

Bei der Lösung von Problemen gibt es auch Fälle, in denen Sie die Reihenfolge für die Berechnung eines Ausdrucks oder den Weg für den Abschluß einer Aufgabe genau festlegen wollen. In diesen Fällen können Sie die Reihenfolge mit den Klammentasten $($ $)$ steuern, die anschließend behandelt werden. Klammern erfordern erhöhte Aufmerksamkeit in der Mathematik – und entsprechend werden sie vom Rechner behandelt.

ABSCHNITT 1:

GRUNDFUNKTIONEN UND TASTEN

() – Klammtasten

In einer Vielzahl von Problemen ist es erforderlich, die genaue Reihenfolge für die Berechnung eines Ausdrucks oder die Gruppierung der Zahlen bei der Problemlösung genau festzulegen. Klammern geben Ihnen eine Möglichkeit, diese Gruppierung von Zahlen und Operationen vorzunehmen. Wenn Sie eine Reihe von Zahlen und Operationen einklammern, weisen Sie damit den Rechner an, diesen Ausdruck zuerst zu bearbeiten, auf eine einzelne Zahl zu reduzieren, und dieses Ergebnis in der restlichen Berechnung zu verwenden. Innerhalb jedes Klammerpaars arbeitet der Rechner nach den Regeln der algebraischen Hierarchie. Setzen Sie auch dann Klammern, wenn Sie Zweifel haben, wie der Rechner einen Ausdruck verarbeiten wird.

Die Anzahl der Klammern, die gleichzeitig offen sein können, ist begrenzt, ebenso die Anzahl der möglichen unvollständigen Operationen. Mit Ihrem Rechner können Sie gleichzeitig bis zu 9 offene Klammern mit bis zu 4 unvollständigen Operationen haben – wenn Sie diese Grenzen überschreiten, blinkt die Anzeige. (Diese Einschränkung wird aber kaum Probleme verursachen.)

Achten Sie auf einen wesentlichen Punkt, wenn Sie Klammern verwenden. Oft sehen Sie Gleichungen oder Ausdrücke, die mit Klammern geschrieben sind, um eine Multiplikation anzudeuten: $(2 + 1)(3 + 2) = 15$. Der Rechner kann keine Multiplikationen erkennen, die auf diese Weise angedeutet sind. Die Operation zwischen den Klammerpaaren muß unbedingt eingegeben werden.

() **2** **+** **1** **)** **×** **(** **3** **+** **2** **)** **=** **15**

Hier ein Beispiel für die Anwendung von Klammern:

Berechnen $\frac{(8 \times 4) + (9 \times -19)}{(3+10-7) \times 2} =$

Lösung: In Aufgaben dieses Typs soll der Rechner zunächst den Zahlenwert des Zählers bestimmen, und dann diesen Wert durch den Nenner dividieren. Um sicherzustellen, daß die Berechnung tatsächlich auf diese Weise stattfindet, setzen Sie bei der Eingabe des Problems Zähler und Nenner extra noch in Klammern.

Tasten	Anzeige	Bemerkungen
((8 × 4)) +	32.	(8×4) wird angezeigt
(9 × 19 + -)		
) +	-139.	Wert des Zählers
((3 + 10) ÷ 7		
) × 2)	8.8571429	Wert des Nenners
=	-15.693548	Endergebnis

ABSCHNITT 1: GRUNDFUNKTIONEN UND TASTEN

Speichertasten

Für die Anwendung stehen Ihnen in Ihrem Rechner 8 Mehrzweck-Speicher zur Verfügung. Diese Speicher sind spezielle Plätze im Gerät, wo Sie Zahlen für die spätere Verwendung speichern können. Die Speicher sind sehr vorteilhaft — in mancher Hinsicht geben sie Ihnen mehrere Rechner in einem, weil Sie damit Zahlen speichern, aufrufen oder arithmetische Berechnungen ohne Einfluß auf die Aufgaben durchführen können, die gerade im Hauptteil des Geräts ablaufen.

Die Tasten **CE** und **CLR** wirken nicht auf die Speicherinhalte — aber sie können mit der Folge **INV** **2nd** **C.t** bei Bedarf vollständig gelöscht werden (ebenso, wenn Sie den Rechner aus- und wieder einschalten.)

Da Sie 8 Speicher haben, müssen Sie den Rechner anweisen, mit welchem Speicher Sie jeweils arbeiten wollen. Wenn Sie also eine Speichertaste drücken, muß unmittelbar danach die Nummer des Speichers angegeben werden ($n = 0,1,2,3,4,5,6$ oder 7). Damit erkennt der Rechner, auf welchen der 8 Speicher Sie sich im Augenblick beziehen. Der logische Aufbau der Operationen der Speichertasten ist nachstehend erläutert.

STO n — Die Speichertaste (Store)

Mit dieser Taste speichern Sie die Zahl in der Anzeige in dem durch n bestimmten Speicher ($n = 0,1,2,3,\dots,7$). (Dabei wird jede zuvor in n gespeicherte Zahl automatisch gelöscht.)

RCL n — Die Aufruftaste (Recall)

Wenn Sie **RCL** n drücken, erscheint die in n gespeicherte Zahl in der Anzeige, und kann in Operationen und Berechnungen verwendet werden. Nach der Anweisung **RCL** n bleibt die Zahl im Speicher erhalten, Sie können also in einer Berechnung den Wert so oft wie nötig aus einem Speicher aufrufen. Eine gespeicherte Zahl bleibt so lange im Speicher erhalten, bis sie durch eine andere Speicheroperation geändert wird oder bis eine Löscheoperation mit den Tasten **INV** **2nd** **C.t** erfolgt oder bis Sie den Rechner ausschalten.

ABSCHNITT 1: GRUNDFUNKTIONEN UND TASTEN

Ein Beispiel für die Anwendung der Speicher:

Vorausgesetzt sei:

$$a = 10.3 \cdot (25 - 1.7)$$

$$b = 15a + 6$$

$c = 20b$, Sie müssen c berechnen.

Tasten	Anzeige	Bemerkungen
10.3 \times 25 -		Berechnung und Speicherung von a
1.7 \times =	239.99	
STO 1	239.99	a wird in Speicher 1 gespeichert
15 \times RCL 1 + 6 =	3605.85	
STO 2	3605.85	b wird in Speicher 2 gespeichert
20 \times RCL 2 =	72117.	Der Wert von c.

Speicherarithmetik

Neben den allgemeinen Speichertasten steht eine Reihe weiterer Tasten zur Verfügung, mit denen die Durchführung von Arithmetik mit gespeicherten Zahlen ohne Einfluß auf laufende Berechnungen möglich ist.

Datamath Calculator Museum

SUM n – Die Taste für Speicheraddition – ermöglicht die direkte algebraische Addition einer Zahl in der Anzeige zu einer gespeicherten Zahl. (Dabei werden keine laufenden Berechnungen beeinflußt.) Das Ergebnis der Addition bleibt im Speicher.

Anmerkung: Dieser Vorgang unterscheidet sich von der Wirkung bei Anwendung der Taste **STO**. Die Operation **STO** n löscht die Zahl im Speicher und ersetzt sie durch die Zahl in der Anzeige. (Die Anzeige wird durch diese Operation nicht beeinflußt.)

INV **SUM** n – Die Tastenfolge für Speichersubtraktion – subtrahiert die Zahl in der Anzeige von der Zahl im Speicher n. Das Ergebnis bleibt in n gespeichert. (Die Anzeige bleibt unverändert.)

2nd **Prd** n – Die Tastenfolge für Speichermultiplikation – multipliziert die Zahl im Speicher n mit dem Anzeigehalt. Das Produkt bleibt in n gespeichert. (Die Anzeige bleibt unverändert.)

INV **2nd** **Prd** n – Tastenfolge für Speicherdivision – dividiert die Zahl im Speicher n durch die Zahl in der Anzeige. Der Quotient bleibt in n gespeichert. (Beachten Sie, daß **INV** und **2nd** in beliebiger Reihenfolge gedrückt werden können, also auch **2nd** **INV** **Prd** n für jede Speicherdivision.) Die Anzeige bleibt bei dieser Operation unverändert.

2nd **Exch** n – Die Tastenfolge für den Austausch vertauscht einfach die Zahl im Speicher gegen die Zahl in der Anzeige. (Der Anzeigewert wird in n gespeichert und der in n gespeicherte Wert wird angezeigt.)

[x:t**] – Die x-t-Austauschtaste**

Diese spezielle Taste vertauscht die Zahl in der Anzeige mit der Zahl im Speicher 7. Speicher 7 wird in verschiedenen Sonderfunktionen verwendet und wird auch als „t“- oder „Test“-Register bezeichnet. Die Taste [**x:t**] ist in ihrer Wirkung identisch mit der Folge [**2nd**] [**Exe**] 7.

Weitere Anmerkungen zu den Speichern

Die 8 Speicher in Ihrem Rechner dienen primär für Ihren Gebrauch und Ihren Vorteil – aber es gibt Fälle (vor allem in komplexen Berechnungen), in denen zusätzlicher Raum für die Verarbeitung benötigt wird. In solchen Aufgaben muß der Rechner auch die Speicher verwenden, und dabei können Zahlen beeinflußt werden, die Sie gespeichert haben. Im folgenden sind diese Fälle angegeben – und sie werden noch einmal bei der Behandlung der verschiedenen speziellen Merkmale des Geräts erläutert.

- Wenn Sie einen komplexen Ausdruck mit drei oder vier Ebenen unvollständiger Operationen berechnen, werden die Speicher 5 und 6 belegt.
- Speicher 7, das t-Register, wird immer dann verwendet, wenn Sie die Taste drücken.
- Statistische Funktionen und Trendlinienanalysen (Details folgen später) können die Speicher 0, 1, 2, 3, 4, 5 und 7 belegen.
- Die Tastenfolge [**2nd**] [**0:t**], die in Kapitel 3 erklärt ist, arbeitet mit Speicher Null.

In der Praxis ergeben sich aus diesen Speichersituationen selten Probleme. Wenn Sie aber den Rechner sehr stark beladen, müssen Sie sich bewußt sein, daß gespeicherte Informationen beeinflußt werden können.

© 2010 Joerg Woerner
Datamath Calculator Museum

ABSCHNITT 2: TECHNISCH-WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

In diesem Abschnitt wird die Erläuterung einiger Eigenschaften Ihres Rechners fortgesetzt, die besonders in Technik, Wissenschaft und höherer Mathematik von Vorteil sind (Anwendungsbereiche, bei denen man bis vor kurzem noch mit mechanischen Rechenschiebern und Tabellen arbeitete.) Viele dieser Merkmale wurden erst durch neue Entwicklungen in der Technologie der integrierten Schaltkreise (IC) möglich. Diese neuen Entwicklungen erlauben, daß verschiedene Funktionen und Tabellen direkt in den IC Rechenkreis eingebracht werden. Auf diese Weise können Sie den Rechner veranlassen, eine komplexe Aufgabe mit einem Tastendruck zu verarbeiten.

EE — Taste für Exponentialform

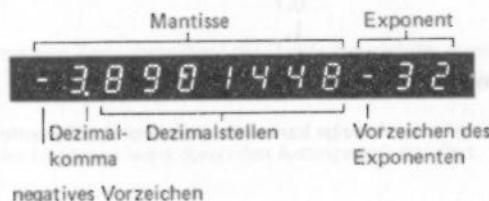
Insbesondere in vielen wissenschaftlichen und technischen Anwendungsbereichen muß man oft mit extrem großen oder kleinen Zahlen rechnen. Solche Zahlen können Sie und Ihr Rechner mühelos mit der Exponentialform verarbeiten. Eine Zahl in Exponentialform wird als eine Basiszahl (oder „Mantisse“), multipliziert mit einer Zehnerpotenz (oder dem „Exponenten“), ausgedrückt.

Mantisse $\times 10^{\text{Potenz}}$

Eingabeverfahren für eine Zahl in Exponentialform:

- Eingabe der Mantisse (dann drücken Sie $+\!-\!$, wenn sie negativ ist).
- Taste EE (Exponenten-Eingabe) — rechts von der Anzeige wird „00“ ausgewiesen.
- Eingabe der Zehnerpotenz (und dann $+\!-\!$, wenn die Zehnerpotenz negativ ist.)

Eine Zahl wie $-3.8901448 \times 10^{-32}$ erscheint wie folgt in der Anzeige:



In der Exponentialform gibt die Zehnerpotenz (die beiden Stellen rechts in der Anzeige) an, wo das Dezimalkomma stehen müßte, wenn Sie die Zahl ausschreiben. Ein positiver Exponent sagt aus, wieviele Stellen das Komma nach rechts verschoben werden müßte, ein negativer Exponent — um wieviele Stellen das Komma nach links rückt. Zum Beispiel: 2.9979×10^8 entspricht 2.99790000.

(Rücken Sie das Komma um 8 Stellen nach rechts und fügen Sie so viele Nullen an wie nötig.)

1.6021×10^{-19} entspricht .000 000 000 000 000 1.6021

(Verschieben des Kommas um 19 Stellen nach links, und Einfügen von so vielen Nullen wie nötig.)

ABSCHNITT 2: TECHNISCH-WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

[2nd] Fix n – Testkomma-Einstellung

Mit dieser vorteilhaften Eigenschaft ist es möglich, bei Ihren Berechnungen die Zahl der angezeigten Stellen rechts vom Dezimalkomma zu wählen. Drücken Sie einfach **[2nd] Fix** und dann die gewünschte Anzahl der Dezimalstellen (0 bis 8). Versuchen Sie es – drücken Sie **[2nd] Fix 3** – und die Anzeige ändert sich sofort auf „0.000“.

Der Rechner rundet alle nachfolgenden Ergebnisse auf diese Dezimalstellenanzahl. Zahleneingaben können jedoch weiterhin mit jeder gewünschten Stellenzahl erfolgen, und auch die 11-stellige interne Genauigkeit des Rechners bleibt erhalten. Der Anzeigewert wird weiter korrekt auf die gewählte Dezimalstellsenzahl gerundet. Beachten Sie auch, daß Sie sowohl in der Standardanzeige als auch in der Exponentialform die gewünschte Anzahl der Dezimalstellen mit der Festkommataste wählen können. Wenn Sie den Rechner aus- und wieder einschalten, löschen Sie die Festkomma-Einstellung. Berechnen Sie folgendes Beispiel:

Beispiel: $\frac{2}{3} = 0.6666667$

© 2010 Joerg Woerner

DataMath Calculator Museum

Tasten	Anzeige/Bemerkungen
AUS-EIN	0
[CLR]	0
2 [−] 3 [=]	0.6666667 Beachten Sie: der Anzeigewert wird korrekt gerundet.
2nd Fix 6	0.666667
2nd Fix 2	0.67
2nd Fix 1	0.7
2nd Fix 0	1.

Löschen der Festkomma-Einstellung

Die Festkomma-Einstellung in der Anzeige kann mit nachstehenden Tastenfolgen aufgehoben werden:

[INV] [2nd] [Fix]

[2nd] [Fix] 9

Kurzes Ausschalten und wieder Einschalten des Rechners (Gesamtlösung).

ABSCHNITT 2: TECHNISCH-WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

2nd **π** – Die Zahl Pi

Die Tastenfolge **2nd** **π** zeigt die ersten acht Stellen der Zahl π an. (Elf Stellen sind im Rechner gespeichert – acht korrekt gerundete Stellen werden angezeigt. – Die angezeigte Zahl lautet: 3.1415927. Die Anzeige von π erfolgt sofort, ohne Einfluß auf laufende Berechnungen. π kann jederzeit in einer Berechnung verwendet werden.

2nd **|x|** – Absolutwert

Drückt man diese Tastenfolge, wird das Vorzeichen der angezeigten Zahl (x) positiv. Diese Eigenschaft findet in einer Vielzahl von Rechen- und Programmsituationen Anwendung.

Wirkung der Tastenfolge **2nd** **|x|**:

Ist die Zahl in der Anzeige negativ, wird sie positiv.

Ist die Zahl in der Anzeige positiv, bleibt sie unverändert.

Sie können diese Funktion jederzeit in einer Berechnung anwenden, sie beeinflußt keine laufenden Berechnungen.

Beachten Sie, daß auf mehreren Tasten der Buchstabe "x" als Teil der Benennung angegeben ist. Dieses "x" bezieht sich auf die angezeigte Zahl.

x² **√x** **1/x** – Quadrat, Quadratwurzel, Reziprokwert

Diese drei Tastenfunktionen sind wesentlich für die schnelle Berechnung einer Vielzahl algebraischer Aufgaben und Gleichungen. Jede dieser Tasten wirkt unmittelbar auf den Anzeigewert x und beeinflußt keine laufenden Berechnungen.

x² – Quadrat-Taste; ermittelt das Quadrat der angezeigten Zahl (Multiplikation der Zahl mit sich selbst).

√x – Quadratwurzel-Taste; berechnet die Quadratwurzel der angezeigten Zahl. Die Quadratwurzel einer Zahl (x) ist die Zahl, die mit sich selbst multipliziert, x ergibt, d.h.:
 $x = (\sqrt{x}) \times (\sqrt{x})$

1/x – Reziprokwert-Taste; 1 wird durch den Anzeigewert dividiert.

ABSCHNITT 2: TECHNISCH-WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

y^x – Allgemeine Potenzen

Mit dieser leistungsfähigen Taste kann man jede (positive) Zahl mit einem Tastendruck zu einer beliebigen Potenz erheben. Die Anwendung:

- Eingabe der Zahl (y), die zu einer Potenz erhoben werden soll.
- Taste **y^x**
- Eingabe der Potenz (x)
- Taste **=** (oder eine andere Operationstaste).

Beispiel: Berechnen Sie $3.1897^{4.7343}$

Tasten	Anzeige/Bemerkungen
CLR	0 Löschen
3.1897 [y^x]	3.1897 y-Wert
4.7343	4.7343 x-Wert
=	242.60674 Endergebnis: y^x

INV y^x – Allgemeine Wurzeln ($\sqrt[x]{y}$)

Mit dieser Tastenfolge kann man die Wurzeln einer beliebigen positiven Zahl berechnen (Umkehrung von y^x - oder $\sqrt[x]{y}$). Ehe die Rechner auf den Markt kamen, waren solche Berechnungen sehr zeitraubend – und man mußte in Logarithmentafeln nachschlagen.

Anwendung dieser Tastenfolge zur Berechnung von $\sqrt[3]{y}$:

- Eingabe der Zahl (y), deren Wurzel zu berechnen ist.
- Tasten **INV** **y^x**
- Eingabe der Wurzel (x)
- Taste **=** (oder eine andere Operationstaste).

Beispiel: Berechnen Sie $\sqrt[3]{21.496}$

Tasten	Anzeige/Bemerkungen
CLR	0 Löschen
21.496 [INV] [y^x]	21.496 y-Wert
3.871	3.871 x-Wert
=	2.2089685 Endergebnis: $(\sqrt[x]{y})$.

Anmerkung: Die Tastenbefehle **y^x** (und **INV** **y^x**) führen Berechnungen aus, die Rechner-intern einige Schritte erfordern. Der Abschluß dieser Berechnungen dauert daher im Vergleich zu anderen etwas länger. Achten Sie darauf, daß der Rechner die Operation beendet – daß ein Endergebnis angezeigt wird, ehe Sie weitere Tasten drücken.

ABSCHNITT 2:

TECHNISCH-WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

[lnx] und **[2nd] [log]** – Logarithmus-Tasten

Logarithmen sind mathematische Funktionen, die in einer großen Anzahl technischer und theoretischer Berechnungen Eingang finden. Daneben bilden die Logarithmen einen wesentlichen Teil der mathematischen Modelle von natürlichen Phänomenen. Die Logarithmen-Tasten ermöglichen Ihnen den unmittelbaren Zugriff auf den Logarithmus einer Zahl – das Nachschlagen in Tabellen erübrigtsich.

[lnx] – natürlicher Logarithmus – der natürliche Logarithmus (Basis e = 2.7182818) der Zahl in der Anzeige wird sofort ausgewiesen. (Anmerkung: Der Anzeigewert muß positiv sein, – beim Versuch, **[lnx]** mit einer negativen Zahl zu berechnen, blinkt die Anzeige.)

[2nd] [log] – Zehnerlogarithmus – der Zehnerlogarithmus (Basis 10) der (positiven) Zahl in der Anzeige wird sofort ausgewiesen.

[INV] [lnx] und **[INV] [2nd] [log]** – Antilogarithmus-Tasten

Diese Tastenfolgen berechnen die „Umkkehrung“ oder den Antilogarithmus (Numerus). ($e^{[lnx]} = x$ und $10^{[\log x]} = x$.) Derartige Berechnungen müssen in vielen technischen Problemen durchgeführt werden. Mit nur einigen Tastenbefehlen auf Ihrem Rechner werden sie schnell gelöst.

[INV] [lnx] – e^x -Tastenfolge – e wird in die Potenz der Zahl in der Anzeige erhoben. (Berechnung des natürlichen Antilogarithmus der ausgewiesenen Zahl.)

[INV] [2nd] [log] – 10^x -Tastenfolge – 10 wird in die Potenz der angezeigten Zahl erhoben. (Berechnung des dekadischen Antilogarithmus des Anzeigewerts.)

Anmerkungen zu den Logarithmus- und Antilogarithmus-Tasten:

Jede dieser Tasten wirkt unmittelbar auf die angezeigte Zahl und beeinflußt keine laufenden Berechnungen. Ihr Rechner verwendet eine Vielzahl von „Routinen“, um diese Werte zu erhalten. Daraus ergeben sich bestimmte Einschränkungen, die zu einer Fehleranzeige führen können, wenn man sie nicht beachtet. Genaue Informationen zu den Fehlerbedingungen siehe Anhang C.

Einige Beispiele: Berechnen Sie $\log 15.32$, $\ln 203.451$, $e^{-0.69315}$, 10^7

Tasten	Anzeige/Bemerkungen
15.32 [2nd] [log]	1.1852588
203.451 [lnx]	5.3154252
.69315 [+/-] [INV] [lnx]	0.4999986
[2nd] [e^x] [2nd] [INV] [log]	1385.4557

ABSCHNITT 2: TECHNISCH-WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

[2nd] **Int** – Ganzzahliger Teil einer Zahl

Diese Tastenfolge weist den ganzzahligen Teil einer Zahl in der Anzeige aus. Wenn Sie zum Beispiel die Zahl 3.1117 eingeben und dann die Tasten **[2nd]** **Int** drücken, entfernt der Rechner alle Stellen rechts vom Dezimalkomma und zeigt nur die ganze Zahl 3 an. (Versuchen Sie es selbst!)

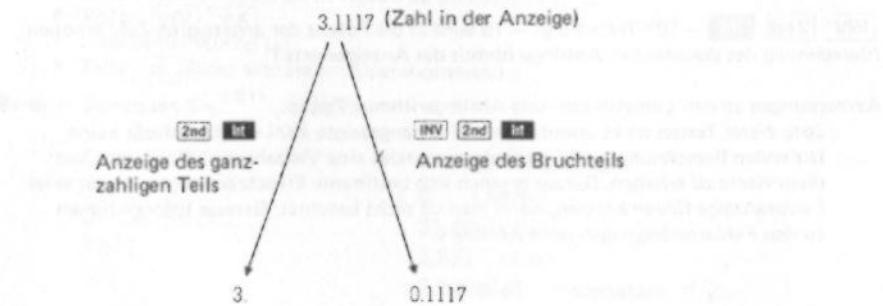
In vielen Fällen in der Mathematik ist es notwendig, nur mit dem ganzzahligen Teil einer Zahl zu rechnen. In der Programmierung erweist sich diese Tastenfolge für die Lösung solcher Situationen als sehr nützlich.

[2nd] **INV** **Int** – Bruchteil einer Zahl

In der Mathematik und in der Programmierung gibt es Fälle, in denen man den Bruchteil einer Zahl isolieren muß und nur mit ihm weiterrechnen kann. Diese Tastenfolge weist den Bruchteil oder die Dezimalstellen einer Zahl in der Anzeige aus. Wenn Sie also 3.1117 eingeben und die Tasten **[2nd]** **INV** **Int** drücken, erscheint 0.1117 in der Anzeige.

Die Skizze zeigt die Wirkung dieser beiden Tastenfolgen.

© 2010 Joerg Woerner
Datamath Calculator Museum



Anmerkung: Die ursprüngliche Zahl geht verloren, wenn Sie **Int** oder **INV** **Int** anwenden – wenn Sie die Zahl später noch benötigen, muß sie zuvor gespeichert werden.

ABSCHNITT 2: TECHNISCH-WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

2nd **Deg** , **2nd** **Rad** , **2nd** **Grad** – Winkeleinheiten

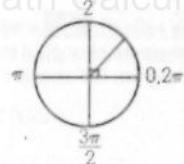
Ihr Rechner kann eine Vielzahl von Aufgaben mit Winkeln ausführen – im wesentlichen trigonometrische Funktionen und polar-/rechteckige Umrechnungen. Mit den nachstehenden Tastenfolgen können Sie bei diesen Berechnungen für die Winkelmessung eine der drei gebräuchlichen Winkeleinheiten wählen.

2nd **Deg** – Einstellung auf (Alt)grad. In diesem Winkelmodus werden alle eingegebenen und berechneten Winkel in Altgrad angegeben, bis Sie einen anderen Modus wählen.
(1 Altgrad entspricht $\frac{1}{360}$ eines Kreises – ein rechter Winkel ist 90°)

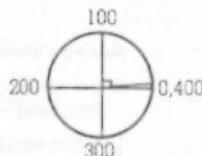


2nd **Rad** – Einstellung auf Radian. Alle Winkel werden in diesem Modus in Radian gemessen.
(1 Radian entspricht $\frac{1}{2\pi}$ eines Kreises, ein rechter Winkel ist $\frac{\pi}{2}$ Radian.)

Datamath Calculator Museum



2nd **Grad** – Einstellung auf Neugrad (Gon). In diesem Modus werden alle Winkel in Gon ausgedrückt. (1 Gon entspricht $\frac{1}{400}$ eines Kreises, ein rechter Winkel ist 100 Gon.)



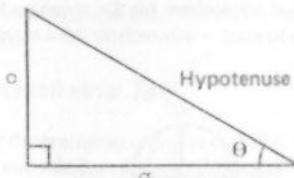
Achtung: Beim Einschalten ist der Rechner immer auf Altgrad eingestellt. Diese Winkeleinheit wird beibehalten, bis Sie einen anderen Modus wählen.

Einer der häufigsten Fehler beim Rechnen mit Winkeln ist die falsche Einstellung des Winkelmodus. Sobald Sie eine Winkeleinheit gewählt haben, wird diese beibehalten, bis Sie selbst wieder eine Änderung vornehmen, oder bis Sie den Rechner aus- und wieder einschalten (beim Einschalten ist der Rechner immer auf Altgrad eingestellt.)

ABSCHNITT 2: TECHNISCH-WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

2nd sin , 2nd cos , 2nd tan – Tasten für Trigonometrische Funktionen

Diese Tastenfolgen berechnen sofort Sinus, Kosinus und Tangens des Winkels in der Anzeige (der Winkel ist in den Einheiten des jeweils gewählten Winkelmodus ausgedrückt). Die folgende Darstellung zeigt, wie die trigonometrischen Funktionen Winkel und Seiten eines rechtwinkligen Dreiecks in Beziehung bringen.



$$\sin \theta = \frac{a}{\text{hyp}} \quad \cos \theta = \frac{c}{\text{hyp}} \quad \tan \theta = \frac{a}{c}$$

INV 2nd sin , INV 2nd cos , INV 2nd tan

Stellt man **INV** vor eine andere Taste, werden Operation und Zweck dieser Taste umgekehrt.

Die Tastenfolgen **INV 2nd sin**, **INV 2nd cos** und **INV 2nd tan** berechnen den Winkel (in Einheiten der gewählten Winkeleinstellung), dessen Sinus, Kosinus oder Tangens angezeigt ist. (Mit den Tastenfolgen berechnen Sie also $\arcsin(\sin^{-1})$, $\arccos(\cos^{-1})$ und $\arctan(\tan^{-1})$.)

Beispiele: Berechnen Sie den Sinus von 90° , 90 Radian und 90 Gon sowie den \arctan von 1.

Tasten	Anzeige/Bemerkungen
AUS-EIN CLR	0 Gesamtlösung (Winkeleinheit Altgrad beim Einschalten)
90 2nd sin	1. Sinus von 90 (Alt)grad
90 2nd sin 2nd sin	0.8939967 Sinus von 90 Radian
90 2nd sin 2nd sin	0.9876883 Sinus von 90 Gon
1 2nd sin INV 2nd tan	45. Der Winkel in Altgrad, dessen Tangens 1 beträgt.

ABSCHNITT 2: TECHNISCH WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

Winkelumrechnungen

Oft ist es erforderlich, die Werte von Winkeln von einer Einheit in eine andere umzurechnen. Für diesen Zweck hat der Rechner zwar keine speziellen Umrechnungstasten, aber die Tastenfolgen zur Umformung von Winkeleinheiten im ersten und vierten Quadranten sind leicht verständlich und können ohne Einfluß auf Speicher oder laufende Berechnungen verwendet werden.

- 1) Achten Sie auf die richtige Einstellung des Winkelmodus und geben Sie den Winkel ein, der umzuformen ist.
- 2) Drücken Sie **2nd sin**
- 3) Wählen Sie die gewünschte Winkeleinheit.
- 4) Drücken Sie **INV 2nd sin**

Beispiel: Umrechnung von 50 Altgrad in Radian

Tasten	Anzeige	Bemerkungen
2nd deg	0	
50 2nd sin	0.7660444	
2nd rad		
INV 2nd sin	0.8726646 Radian	

Der Bereich der Winkel für diese Umrechnungen ist auf den ersten und vierten Quadranten beschränkt. Bei größeren Winkeln können Sie direkt umrechnen:

- Altgrad $\times \frac{100}{90} =$ Neugrad (Gon)
- Altgrad $\times \frac{\pi}{180} =$ Radian
- Neugrad $\times \frac{90}{100} =$ Altgrad
- Neugrad $\times \frac{\pi}{200} =$ Radian
- Radian $\times \frac{180}{\pi} =$ Altgrad
- Radian $\times \frac{200}{\pi} =$ Neugrad

ABSCHNITT 2: TECHNISCH WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

Umrechnungen

Ihr Rechner verfügt über zwei Tasten, die sich vor allem bei Umrechnungen als zweckmäßig erweisen.

2nd D.MS – (Alt)Grad-Minuten-Sekunden in Dezimalgrad

Diese Tastenfolge rechnet einen Winkel, der in Grad-Minuten-Sekunden ausgedrückt ist, in Dezimalgrad um. Die entgegengesetzte Rechnung – Umformung von Dezimalgrad in Grad-Minuten-Sekunden – erfolgt über die Tasten **INV 2nd D.MS**. Diese Umrechnung eignet sich für die Durchführung von Arithmetik mit Winkeln und Zeitangaben – ausgedrückt in Stunden-Minuten-Sekunden.

Bei Eingabe und Umrechnung eines Winkels achten Sie auf die Form, die aus der Beschriftung der Taste **D.MS** bereits hervorgeht:

- Eingabe der Gradzahl
- Dezimalkomma-Taste **.**
- Eingabe der Anzahl der Minuten (2 Stellen – kleiner als 60)
- Eingabe der Anzahl der Sekunden (2 Stellen – kleiner als 60)
- Tastenfolge **2nd D.MS**

Die dezimale Entsprechung des Winkels oder der Zeit wird jetzt angezeigt. Ebenso mühelos ist die Umkehrung:

- Eingabe des Winkels (ganze Zahl und Dezimalstellen)
- Tastenfolge **INV 2nd D.MS**

Der Winkel wird nun in folgender Form ausgewiesen: Gradzahl-Dezimalkomma-Minuten-Sekunden.

Beispiel: Rechnen Sie $47^{\circ}05'38''$ in Dezimalgrad und dann wieder in Grad - Minuten - Sekunden um.

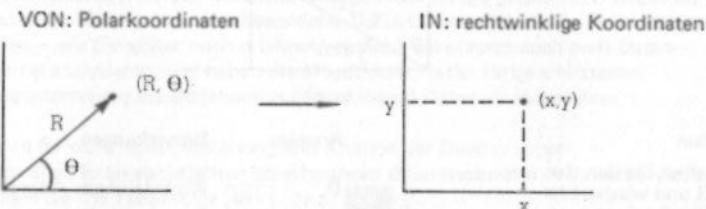
Tasten	Anzeige	Bemerkungen
47.0538 2nd D.MS	47.093889	dezimale Entsprechung des Winkels
INV 2nd D.MS	47.0538	wieder Grad-Minuten-Sekunden

Beachten Sie, daß Sie auf dieselbe Weise auch Stunden-Minuten- Sekunden in die entsprechende Dezimalform umrechnen können.

ABSCHNITT 2: TECHNISCH WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

2nd P-R – Polar/Rechtwinklige Umrechnungen

Diese zweckmäßige Eigenschaft Ihres Rechners ist vor allem in wissenschaftlichen und technischen Anwendungsbereichen von Vorteil. Durch Anwendung der Taste **[xcst]** ist es schnell und einfach, Polarkoordinaten in rechtwinklige Koordinaten umzuformen und umgekehrt. Beachten Sie bei der Umrechnung die unten angegebenen Tastenfolgen:



Umrechnung von Polarkoordinaten in rechtwinklige Koordinaten:

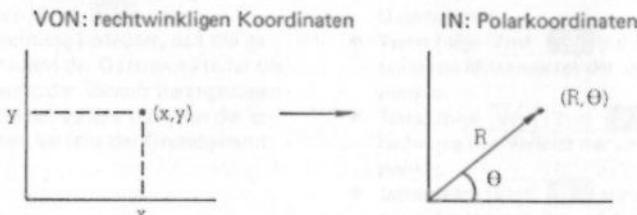
- Eingabe des Wertes für „R“
- Taste **[xcst]**
- Eingabe des Wertes für „θ“ (in den mit den Winkelmodus-Tasten gewählten Einheiten)
- Tastenfolge **2nd P-R**

„y“ wird jetzt angezeigt. Um den Wert für „x“ abzulesen:

- Drücken Sie **[xcst]**

Damit wird „x“ angezeigt.

Umrechnung von rechtwinkligen Koordinaten in Polarkoordinaten:



- Eingabe des „x“-Wertes
- Taste **[xcst]**
- Eingabe des „y“-Wertes
- Tastenfolge **2nd P-R**

„θ“ wird jetzt angezeigt (in der gewählten Winkeleinheit).

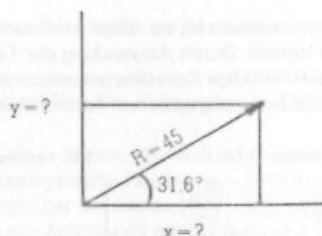
Um den Wert für „R“ abzulesen, drücken Sie:

- Taste **[xcst]**

Damit wird „R“ angezeigt.

ABSCHNITT 2: TECHNISCH WISSENSCHAFTLICHE FUNKTIONEN UND TASTEN

Beispiel: Rechnen Sie $R = 45$ Meter, $\theta = 31.6^\circ$ in rechtwinklige Koordinaten um.



Tasten

Schalten Sie den Rechner
AUS und wieder EIN
 $45 \text{ [x]} 31.6$

[**2nd**] [**F4**]
[**x²**]

Anzeige

0

23.579366
38.327712

Bemerkungen

Winkelheit: Altgrad

Eingabe der Polarkoordinaten: R ,
dann θ

„y“-Wert
„x“-Wert

Anmerkung: Bei dieser Umrechnung wird intern Speicher 7 (das t-Register) belegt. Verwenden Sie also nicht Speicher 7, wenn Sie polar/rechtwinklige Umrechnungen durchführen.
(Zahlen, die Sie hier gespeichert haben, gehen verloren.)

ABSCHNITT 3: STATISTISCHE FUNKTIONEN UND TASTEN

In vielen Situationen – ob in Schule, Alltag oder Beruf müssen oft große Gruppen von Datenpunkten verarbeitet werden. Diese Daten können Labor-Versuchsmessungen, Testpunkte oder Ergebnisse von Studien- oder Forschungsprojekten sein, etc.

Ihr Rechner verfügt über Eigenschaften, mit denen es möglich ist, durch Berechnung von Mittelwert, Varianz und Standardabweichung die Daten schnell zusammenzustellen und zu analysieren. In den Speichern werden die Zwischenergebnisse für Sie gespeichert, wobei Sie leicht Zugriff auf diese Werte haben, (ein besonderer Vorteil, wenn statistische Probleme programmiert werden – wie Sie später noch erfahren werden.) Sie können auch zwei Datengruppen gleichzeitig analysieren, und haben die Möglichkeit, in der fortgeschrittenen statistischen Programmierung die Beziehung zwischen diesen Daten zu überprüfen.

Hier das Verfahren für die Zusammenstellung und Analyse der Datengruppen:

- Schalten Sie vor allen Ihren statistischen Berechnungen Ihren Rechner kurz aus und wieder ein oder verwenden Sie die Tastenfolge **INV** **2nd** **C.t**

Analyse einer Datengruppe:

- Eingabe jedes Datenpunkts
- Tasten **2nd** **$\Sigma+$**
- Wiederholung für alle Datenpunkte
- Berechnung des Mittelwertes der Daten über die Tasten **2nd** **\bar{x}** .
- Berechnung der Varianz der Daten (mit N-Gewichtung) über die Tasten **2nd** **σ^2** .
- Berechnung der Standardabweichung der Daten (mit N-Gewichtung) über die Tasten **2nd** **σ** .
- (N-Gewichtung bedeutet, daß die gesamte Anzahl der Datenpunkte für die Berechnung der Varianz herangezogen wird. Hierbei spricht man von der so genannten Varianz der Grundgesamtheit.)

Analyse von zwei Datengruppen gleichzeitig:

Benennen Sie die beiden Datengruppen als (unabhängiges) „x“-Feld und als (abhängiges) „y“-Feld.

- Eingabe eines „x“-Datenpunkts
- Taste **x:t**
- Eingabe eines „y“-Datenpunkts
- Tasten **2nd** **$\Sigma-$**
- Wiederholung für alle Punkte
- Tastenfolge **INV** **2nd** **\bar{x}** zur Berechnung des Mittelwertes der „x“-Datenpunkte
- Tastenfolge **2nd** **\bar{y}** zur Berechnung des Mittelwertes der „y“-Datenpunkte.
- Tastenfolge **INV** **2nd** **σ^2** zur Berechnung der Varianz der „x“-Datenpunkte.
- Tastenfolge **2nd** **σ^2** zur Berechnung der Varianz der „y“-Datenpunkte.
- Tastenfolge **INV** **2nd** **σ** **\sqrt{x}** zur Berechnung der Standardabweichung der „x“-Datenpunkte.
- Tastenfolge **2nd** **σ** **\sqrt{y}** zur Berechnung der Standardabweichung der „y“-Datenpunkte.

STATISTISCHE FUNKTIONEN UND TASTEN

Während Sie Ihre statistischen Analysen durchführen, werden die Daten in den Speichern wie folgt zusammengestellt:

0	n	(die Anzahl der Datenpunkte)
1	Σy	
Speicher	Σy^2	
2	Σx	
3	Σx^2	
4	Σxy	
5		
7		Der zuletzt eingegebene x-Wert plus 1.

Die Formel für die Berechnung der Varianz lautet:

$$\text{Varianz} = \frac{\sum(x_i - \bar{x})^2}{N} \text{ oder } \frac{\sum(y_i - \bar{y})^2}{N}$$

Das Symbol Sigma (Σ) bedeutet „die Summe von“.

Wenn Sie bei der Eingabe eines Datenpunkts einen Fehler machen, können Sie mit folgendem Verfahren korrigieren: Sie geben die falschen Datenpunkte wie zuvor erneut ein, drücken aber bei der Eingabe vor **2nd** **$\Sigma+$** die Taste **INV**.

Beispiel: Sie werten eine Prüfung aus und Sie haben die Punktzahlen schon eingegeben. Sie wollen sehen, wie gut die Klasse abgeschnitten hat. Die Punktzahlen sind nachstehend aufgelistet:

96 65 81

85 76 86

57 98 75

78 100 72

81 70 80

PRESS

INV **2nd** **Σ**

Anzeige/Bemerkungen

0 Gesamtlösung

96 **2nd** **$\Sigma+$**

1. Der Rechner zählt die Datenpunkte

85 **2nd** **$\Sigma+$**

2. für Sie

Wiederholung für alle
Punkte

•

72 **2nd** **$\Sigma+$**

14.

80 **2nd** **$\Sigma+$**

15.

2nd **Σ**

80. Durchschnitt der Klasse

2nd **σ^2** **F6**

11.564313 Standardabweichung

Einige Anmerkungen zu den statistischen Funktionen

Wenn Sie eine Analyse zweier Datengruppen erstellen, wobei die Daten in einer Beziehung zueinander stehen, müssen Sie gelegentlich für jede Eingabe der „y“-Variablen die Variable x um 1 erhöhen. (Zum Beispiel bestimmte Jahreswerte – 1961, 1962, 1963, etc.) In diesem Fall müssen Sie nur den ersten x-Wert eingeben, und der Rechner erhöht für jeden eingegebenen y-Wert den entsprechenden x-Wert automatisch um 1. Diese Eigenschaft ist bei der Programmierung von sogenannten Trendanalysen von besonderem Nutzen. Bei den Berechnungen der Varianz wird grundsätzlich mit N-Gewichtung gearbeitet – die, wie Statistiker wissen, für die Analyse von Grundgesamtheiten angebracht ist. Um Varianzberechnungen mit (N-1)-Gewichtung zu erhalten, können Sie nach der Berechnung der Varianz über die Tastatur die nachstehende Folge eingeben:

X RCL 0 ÷ (RCL 0 - 1 =

Damit multiplizieren Sie mit $\frac{n}{n-1}$

Statistiker verwenden (N-1)-Gewichtung für die Analyse von Stichproben.
(Beachten Sie: die Quadratwurzel der Varianz ist die Standardabweichung.)

© 2010 Joerg Woerner

Datamath Calculator Museum

Ihr Rechner ist ein leistungsfähiges Problemlösungsgerät, das mit einer Vielzahl von sofort anwendbaren Funktionen ausgestattet ist – direkt über die Tastatur. Dazu kommt die Programmierbarkeit. Das heißt, Sie können Ihren Rechner anweisen, „seine eigenen Tasten zu drücken“, um automatisch eine große Menge von Berechnungen durchzuführen, (einschließlich solcher Aufgaben, die Entscheidungen erfordern). Sie können Ihrem Rechner bis zu 50 Schritte zur Problemlösung „lehren“, und ihn dann veranlassen, diese Schritte, so oft Sie es wünschen, durchzuführen – mit einem einzigen Tastendruck! Wenn Sie die erste Hälfte dieser Anleitung schon gelesen haben, sahen Sie auch die Wirkung der Programmierbarkeit. Wenn Sie das Buch nicht von Anfang an kennen, und Sie hier Ihre ersten Programmiererfahrungen machen, empfehlen wir, die Kapitel 3 und 4 zu lesen, um dabei schrittweise die Grundlagen der Programmierung kennenzulernen. Diese Kapitel sind als Kurzlehrgang gedacht, wobei Kapitel 3 mit einer Beschreibung der Tasten fortgesetzt wird, die in Zusammenhang mit Programmen und Programmierung stehen. Wenn Sie bereits Programmiererfahrung haben, sind Sie vielleicht in der Lage, nach der Lektüre dieser Tastenbeschreibung den Rechner zu nehmen und sofort zu programmieren.

Was geschieht intern?

Die Art, wie Ihr Rechner ein Programm „erlernt“, ist in Wirklichkeit sehr einfach. Es gibt einen speziellen Speicher, den „Programmspeicher“, der die Tastenbefehle behält, die Sie anwiesen. Wenn Sie den Rechner programmieren, wird jede eingegebene Tastenfolge der Reihe nach in diesem Speicher als einfacher Kode gespeichert. Sobald die Eingabe des Programms abgeschlossen ist, kann der Rechner zurückgehen, die Kodes nacheinander lesen, und an Ihrer Stelle in der exakten Reihenfolge Ihrer Anweisungen seine eigenen Tasten drücken.

Zur Eingabe eines Programms verwenden Sie die Programmiertasten, die links auf der Tastatur angeordnet sind. Mit diesen Tasten können Sie den Rechner veranlassen, die nötigen Tastenbefehle für die Lösung des Problems aufzunehmen. Wenn Sie Ihrem Rechner ein Programm eingegeben haben, ist dieser Vorgang mit dem Anlegen von Gleisen vergleichbar, auf denen er fahren muß – ein Weg für den Rechner, der bei der Lösung Ihres Problems einzuhalten ist. Ist der Rechner programmiert und vorbereitet, können Sie ihn am Anfang dieser „Gleise“ starten und ihn wie einen Zug die Strecke der eingegebenen Schritte abfahren lassen. Die Strecke ist Ihr Programm, das im Programmspeicher gespeichert ist, und der Zug ist der sogenannte Programmzeiger. Wenn der Zug auf der Programmstrecke an einem von Ihnen programmierten Schritt vorbeikommt (oder auf diesen Schritt zeigt), drückt der Rechner an Ihrer Stelle exakt nach Anweisung seine eigenen Tasten.

Dieses Kapitel gibt Ihnen einen kurzen Überblick, „wie die Gleise anzulegen sind“ – d.h. über die Programmiertasten Ihres Rechners.

[LRN] — DIE LEARN-TASTE

[RST] — DIE RESET-TASTE

[LRN] — Die Taste Learn

Wenn man diese Taste einmal drückt, schaltet sich Ihr Rechner in den Learn-Modus – und ist damit bereit, ein Programm aufzunehmen. Im Learn-Modus hat die Anzeige ein völlig anderes Format und zeigt 4 (manchmal 5) Ziffern an.

00 00

Die beiden linken Stellen in der Anzeige geben die Nummer des Schrittes an, den Sie als nächsten eingeben. (Sie können bis zu 50 Schritte, 00 bis 49, eintasten.) Die Stellen rechts stehen für die Tastenkodes, zwei- oder drei-stellige Zahlen, die den Rechner anweisen, welche Programmschritte Sie gerade eingeben. Diese beiden rechten Stellen (bei bestimmten Tastenbefehlen kommt noch eine dritte dazu) sind während der Eingabe Ihres Programms Null. Der Grund ist, daß in der Anzeige immer der Schritt ausgewiesen wird, den Sie als nächsten eingeben, und unmittelbar nach der Eingabe wird der nächste freie Platz im Programmspeicher angezeigt. (Später wird erläutert, wie Sie zurückgehen und die Tastenkodes für jeden eingegebenen Schritt überprüfen können.)

Drückt man die Taste [LRN] ein zweites Mal, wird der Learn-Modus aufgehoben, und der Rechner ist für den Ablauf eines Programms oder für manuelle Berechnungen über die Tastatur bereit. An dieser Stelle schaltet sich auch die Anzeige wieder auf ihr Standardformat.

Beachten Sie, daß im Learn-Modus keine Berechnungen über die Tastatur möglich sind. Der Rechner interpretiert jeden eingegebenen Tastenbefehl als Programmschritt, der gespeichert werden muß. Sobald Sie das Programm gespeichert haben, und der Learn-Modus aufgehoben ist, können Sie alle erforderlichen Berechnungen über die Tastatur durchführen – ohne Einfluß auf das Programm im Programmspeicher. Bedenken Sie auch, daß beim Ausschalten des Rechners der Programmspeicher und auch alle anderen Speicher und Register vollständig gelöscht werden.

[RST] — Die Taste Reset (Rückstellen)

Dies ist die Taste für die Rückstellung an den Anfang. Sie weist den Programmzeiger an, zu Schritt 00 zurückzugehen. Sie können die Taste als Teil eines Programms (den Befehl im Learn-Modus) eingeben, oder direkt über die Tastatur im Rechenmodus anwenden. Als Teil eines Programms stellt sich bei diesem Befehl der Programmzeiger sofort auf den ersten Schritt, 00, ein, und die Verarbeitung wird fortgesetzt. Gibt man die Anweisung über die Tastatur, wenn der Learn-Modus aufgehoben ist, wird der Programmzeiger nur auf Schritt 00 gestellt.

[RST] hat noch weitere besondere Funktionen – Nullstellung des Zählers für die Unterprogramm-Ebenen und Löschen des Unterprogramm-Rücksprungregisters.

2nd Pause – PAUSENTASTE

R/S – Die Run/Stop-Taste

Drückt man die Taste **R/S**, (wenn der Learn-Modus aufgehoben ist) wird der Rechner angewiesen, den Programmablauf ab der augenblicklichen Position des Programmzeigers zu beginnen. Drückt man diese Taste (und hält sie kurz fest), wenn das Programm gerade läuft, wird es unterbrochen. Die Taste **R/S** erfüllt zwei Funktionen – ist das Programm unterbrochen, wird der Ablauf wieder aufgenommen; wenn das Programm läuft, wird es unterbrochen.

Die **R/S** Taste kann auch Teil eines Programms sein – wenn man sie während des Learn-Modus verwendet. Wenn das Programm beim Ablauf einen **R/S**-Befehl als Programm schritt erreicht, wird es unterbrochen und der augenblickliche Wert im Anzeigeregister wird ausgewiesen. Auf diese Weise können Sie ein Programm stoppen, um Zwischenergebnisse zu erhalten, oder um neue Zahlen einzugeben, die für den weiteren Ablauf erforderlich sind, oder um zu halten, wenn das Programm beendet ist.

Beachten Sie, daß der Programmzeiger ab dem augenblicklichen Speicherplatz den Programmspeicher durchläuft, wenn Sie die Taste **R/S** im Rechenmodus drücken. Wenn Sie mit dem ersten Programmschritt (Schritt 00) beginnen wollen, müssen Sie vor **R/S** die Taste **RST** drücken. (Mit der später noch erläuterten **GTO**-Taste können Sie den Programmzeiger auf jeden Speicherplatz einstellen, mit dem Sie beginnen wollen.)

2nd Pause – Die Pausen-Taste

Diese Tastenfolge soll als Teil von Programmen verwendet werden, und sie wird während des Learn-Modus ins Programm eingegeben. Wenn der Programmzeiger beim Programmablauf einen **2nd** **Pause**-Befehl erreicht, wird das Programm etwa eine 3/4 Sekunde lang unterbrochen, und der Inhalt des Anzeigeregisters an diesem Punkt wird ausgewiesen. Dann setzt das Programm den Ablauf mit dem nächsten Schritt fort.

Die Tastenfolge [2nd] **Pause** erlaubt einen schnellen Blick auf ein Programmergebnis, ohne die Verarbeitung zu unterbrechen. Dieser Befehl ist vor allem bei ständig wiederkehrenden Programmfolgen (oder „Schleifen“) sehr zweckmäßig – weil Sie die wechselnden Ergebnisse der fortgesetzten Berechnungen gut beobachten können. Sie können sehen, wie sich ein Geldbetrag in einer Zinseszins-Situation im Verlauf der Monate entwickelt, oder die Annäherung einer Funktion an einen Grenzwert verfolgen etc. Wenn die Pause länger als eine 3/4 Sekunde dauern soll, können Sie weitere [2nd] **Pause**-Befehle anfügen. Mit jedem Pausenbefehl wird der Anzeigehinheit um eine 3/4 Sekunde länger ausgewiesen. (Bedenken Sie aber, daß jede zusätzliche [2nd] **Pause**-Tastenfolge einen Ihrer 50 möglichen Programmschritte belegt.)

KOMBINIERTE KODES

2nd Lbl n – Die Label-Tastenfolge

Diese Taste ermöglicht die Markierung jeder Stelle in einem Programm (oder eines ganzen Programmteils), oder eines Unterprogramms (Unterprogramme werden später ausführlich behandelt). Im Learn-Modus drücken Sie die Tastenfolge 2nd Lbl n, wobei n eine Ziffer im Bereich 0 bis 9 ist. Damit können Sie bis zu 10 Punkte in einem Programm kennzeichnen. (Labelnummern dürfen sich im gleichen Programm nicht wiederholen.)

Labels in einem Programm sind sehr vorteilhaft, weil sie mit anderen Tasten (GTO und SBR) zusammenwirken, um den Programmzeiger auf jede beliebige Stelle einstellen zu können. Labels haben keine Verbindung zu anderen Programmschritten oder laufenden Berechnungen – sie dienen lediglich als „Wegweiser“ für Sie und für den Programmzeiger. Sie sollten jede Schlüsselstelle Ihres Programms mit einem Label markieren – besonders, wenn Sie veranlassen wollen, daß sich der Programmzeiger während des Programmablaufs auf diese Stelle richtet, (bzw. zu dieser Stelle „ verzweigt“). Wenn Sie einen Punkt im Programm markieren, können Sie mit der Taste GTO von dieser Stelle an den Programmablauf starten.

Kombinierte Kodes

Die gesamte Tastenfolge 2nd Lbl n belegt nur einen Programmschritt, weil der Rechner den Labelbefehl und die Labelnummer zusammen in einem besonderen Kurzkode, dem so genannten Kombikode, speichert. Prüfen Sie dies selbst mit Ihrem Rechner. Schalten Sie ihn aus und wieder ein, drücken Sie LRN, und dann 2nd Lbl. In der Anzeige erscheint:

00 86 0

Die beiden ersten Stellen weisen auf den Schritt 00 hin. Die beiden nächsten Ziffern sind der Labelkode, 86, und die letzte Stelle (Null) ist der Platz für die Labelnummer (0-9). Um den Labelbefehl zu vervollständigen, geben Sie jetzt eine Labelnummer ein – zum Beispiel 3. Wenn Sie die Taste 3 drücken, ändert sich die Anzeige auf 01 00, d.h. 01 ist der nächste freie Speicherplatz für die Eingabe des Programms.

Mehrere Tastenfolgen im Programm verwenden diese kombinierten Kodes. Grundsätzlich werden die Kodes bei all den Funktionen zusammengefaßt, für deren Vollständigkeit eine Labelnummer oder ein Speicherplatz angegeben werden muß. (Beispiele hierfür sind alle Speicherbefehle wie STO, RCL etc.) Immer, wenn während der Programmierung im Learn-Modus die dritte Stelle ganz rechts aufleuchtet, benötigt der Rechner eine Label- oder Speichernummer (0-9). (Vergessen Sie nicht, eine Nummer einzugeben – und notieren Sie auch, welches Label oder welcher Speicher für welchen Zweck verwendet wird.) Beachten Sie, daß auch der Kode der Tastenfolge 2nd Fix n kombiniert ist.

GTO n – Der Sprungbefehl zum Label („go to“)

Diese Tastenfolge kann Teil eines Programms sein (wenn sie im Learn-Modus gedrückt wird), die Anweisung ist aber auch direkt über die Tastatur möglich, (wenn der Learn-Modus aufgehoben ist). Drückt man **GTO n**, wobei n die Labelnummer (0 bis 9) ist, stellt sich der Programmzeiger sofort auf das Label Nummer n ein. Wenn der Zeiger beim Programmablauf einen **GTO n** Befehl erreicht, richtet er sich sofort auf Label n und setzt das Programm an dieser Stelle fort. (Die Folge **GTO n** wird als ein Programmschritt mit einem kombinierten Kode gespeichert.)

Wenn man die Taste **GTO** in einem Programm verwendet, bewirkt sie einen sogenannten unbedingten Sprung. Wenn der Programmzeiger ein **GTO n** erreicht, stellt er sich sofort auf Label n ein und setzt den Ablauf fort, unabhängig vom jeweiligen Status. Durch Kombination der Folge **GTO n** mit anderen Tasten sind noch weitere Arten von Programmsprüngen (oder Verzweigungen) möglich. Mit diesen anderen Tasten kann man vor dem Befehl **GTO n** eine Bedingung einbringen. Sie haben die Möglichkeit, mit diesen Tasten das Programm so aufzubauen, daß an bestimmten Punkten Entscheidungen getroffen werden – und bedingte Sprünge (bedingte Verzweigungen) programmieren.

Die Folge **GTO n** ist in einer Vielzahl von Situationen sehr vorteilhaft. Sie können zum Beispiel ein langes Programm schreiben, das mehrere verschiedene Funktionen durchführt. Wenn Sie jede Funktion mit einer Labelnummer markieren, können Sie problemlos jede Funktion einzeln anwenden – ohne das gesamte Programm ablaufen zu lassen. Wenn der Learn-Modus aufgehoben ist, drücken Sie nur **GTO n** (die Labelnummer der gewünschten Funktion), und starten das Programm mit der Taste **R/S**. Der Programmzeiger beginnt dann bei der markierten Stelle.

GTO 2nd nn – Der Sprungbefehl zu einer Schrittnummer

Diese Tastenfolge ist nur im Rechenmodus (und nicht im Learn-Modus) möglich. Sie können damit auf einfache Weise den Programmzeiger auf jede beliebige Schrittnummer einstellen. Wenn der Learn-Modus aufgehoben ist, und Sie drücken **GTO 2nd** und eine zweistellige Programmschritt-Nr. (00 bis 49), richtet sich der Programmzeiger sofort auf den Schritt mit dieser Nummer. Diese Eigenschaft eignet sich vor allem für die Änderung oder Korrektur eines Programmschritts (mehr Informationen – siehe Redigiertasten), oder wenn der Programmzeiger an einem nicht markierten Punkt im Programm starten soll.

Vollziehen Sie das mit Ihrem Rechner nach. Schalten Sie ihn aus, dann ein, und drücken Sie **GTO 2nd 16**. Dann schalten Sie mit **LRN** in den Learn-Modus. In der Anzeige erscheint 16 00, d.h., Sie haben den Programmzeiger auf den Schritt Nummer 16 eingestellt. Entsprechend können Sie den Programmzeiger auf jeden anderen Schritt (0 bis 49) richten. (Der Versuch, über Schritt 49 hinauszugehen, führt zu einer blinkenden Fehleranzeige.)

2nd Dsz – DEKREMENT
UND ÜBERSPRINGEN BEI NULL

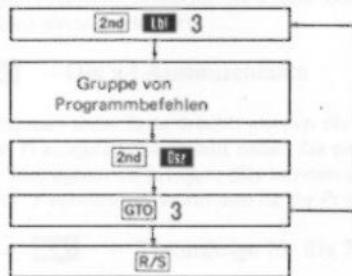
2nd Dsz – Dekrement und Überspringen bei Null

Diese Tastenfolge ist sehr leistungsfähig. Sie können damit eine wiederkehrende Berechnung (oder „Schleife“) bilden, die nach Ihrer Wahl beliebig oft durchläuft. (Sie speichern die Anzahl der Wiederholungen im Speicher Null.) Die tatsächliche Funktion dieser Tastenfolge: Sie haben eine Steuermöglichkeit, ob der Programmzeiger zu dem Schritt unmittelbar nach diesem Befehl weiterrückt, oder ob er diesen Befehl überspringt und das Programm mit dem nächsten Schritt fortsetzt. Wenn der Programmzeiger während des Ablaufs die Tastenfolge

2nd Dsz erreicht, findet nachstehender Vorgang statt:

- Zuerst wird der Inhalt von Speicher Null um 1 vermindert. (Dies setzt voraus, daß der Inhalt von Speicher 0 eine positive, ganze Zahl ist. Ist Speicher Null mit einer gebrochenen Zahl belegt, nimmt der Rechner die nächst höhere, ganze Zahl an. Wenn eine negative Zahl in Speicher 0 ist, wird sie um 1 erhöht.)
- Der Rechner fragt dann: Ist der Inhalt von Speicher Null gleich Null?
- Ist die Antwort nein, rückt der Programmzeiger zu dem Schritt unmittelbar nach **2nd Dsz** weiter.
- Ist die Antwort ja, überspringt der Programmzeiger den Befehl nach **2nd Dsz** und fährt fort.

Nach der Tastenfolge **2nd Dsz** kann ein beliebiger Befehl stehen, oft ist jedoch an dieser Stelle ein **GTO n** Befehl angebracht, um wiederkehrende Berechnungen zu programmieren. In dem unten skizzierten Programm wird zum Beispiel die „Gruppe von Programmbefehl“ wegen der Schleife mehrmals wiederholt.



Wie oft die Gruppe von Programmbefehl durchzuführen ist, wird durch die Zahl im Speicher Null gesteuert (die Zahl wurde vor Ablauf des Programms gespeichert). Wenn im Speicher 0 zum Beispiel die Zahl 6 gespeichert ist, wird die Gruppe von Programmbefehl 6 mal durchlaufen, dann unterbricht das Programm.

Beachten Sie, daß diese Befehlsgruppe auch dann 6 mal durchgeführt wird, wenn 5.2, -6 oder -5.2 der Inhalt von Speicher Null wäre, ehe der Programmzeiger diese Befehlsgruppe erreicht. Da immer bei Anwendung von **2nd Dsz** Speicher Null benötigt wird, kann während der Verarbeitung einer **2nd Dsz**-Folge der Speicher Null nicht für andere Zwecke belegt werden, Sie können aber die Zahl in Speicher Null durch Aufruf als Teil einer Berechnung verwenden.

**[INV] [2nd] [Dsz] – DEKREMENT UND
ÜBERSPRINGEN BEI NICHT-NULL**

Die Tastenfolge **[INV] [2nd] [Dsz]** führt zu einer Tabelle mit den entsprechenden Befehlen.

DEKREMENTO – ZUR VORHERgehende Befehle
ZUM VORHERgehende Befehle

[INV] [2nd] [Dsz] – Dekrement und Überspringen bei Nicht-Null

Diese Tastenfolge wirkt in gleicher Weise wie die vorhergehende, nur mit umgekehrten Bedingungen. Folgender Vorgang findet statt, wenn der Programmzeiger **[INV] [2nd] [Dsz]** erreicht:

- 1 wird vom Inhalt von Speicher Null subtrahiert. (Wiederum wird angenommen, daß der Inhalt von Speicher Null eine positive ganze Zahl ist. Ist eine gebrochene Zahl im Speicher Null, arbeitet der Rechner mit der nächst höheren ganzen Zahl. Ist die Zahl im Speicher 0 negativ, wird 1 addiert.)
- Der Rechner fragt dann: Ist der Inhalt von Speicher Null gleich Null?
- Ist die Antwort nein, überspringt der Programmzeiger den Befehl unmittelbar nach der Folge **[INV] [2nd] [Dsz]**.
- Ist die Antwort ja, rückt der Programmzeiger zu dem Befehl unmittelbar nach **[INV] [2nd] [Dsz]** weiter.

Auch dieser Befehl eignet sich für die Programmierung wiederkehrender Berechnungen in einer Vielzahl von Programmsituationen.

© 2010 Joerg Woerner
Datamath Calculator Museum

Einige Befehle Ihres Rechners sollen Ihnen eine Möglichkeit geben, Entscheidungsprozesse in Ihre Programme einzubeziehen. Die Grundoperation dieser Befehle ist immer ähnlich – sie sind leicht verständlich, aber auch flexibel genug, um ein hohes Maß an Programmierleistung zu bieten.

Wenn eine Tastenfolge für einen bedingten Sprungbefehl in einem Programm erreicht wird, vergleicht der Rechner grundsätzlich zwei Größen: die Zahl x im Anzeigeregister und die Zahl t in einem speziellen Testregister (oder „ t “-Register). (Das Test- oder „ t “-Register ist der Speicher 7 Ihres Rechners.) Der Rechner stellt dann eine der vier Fragen zum Vergleich von x und t – die von dem verwendeten Programmbeispiel abhängen:

2nd	x=t	Ist $x = t$? (Ist x gleich t ?)	
INV	2nd	x=t	Ist $x \neq t$? (Ist x ungleich t ?)
2nd	x=t	Ist $x \geq t$? (Ist x größer oder gleich t ?)	
INV	2nd	x=t	Ist $x < t$? (Ist x kleiner t ?)

Wenn die Frage mit ja beantwortet wird, rückt der Programmzeiger zu dem unmittelbar nachfolgenden Befehl. Ist die Antwort NEIN, überspringt der Programmzeiger den Befehl direkt nach dem Vergleich und führt fort.

In den folgenden Absätzen wird jeder bedingte Sprungbefehl (mit den entsprechenden Tasten) einzeln erläutert.

x=t – Die x - t -Austauschtaste

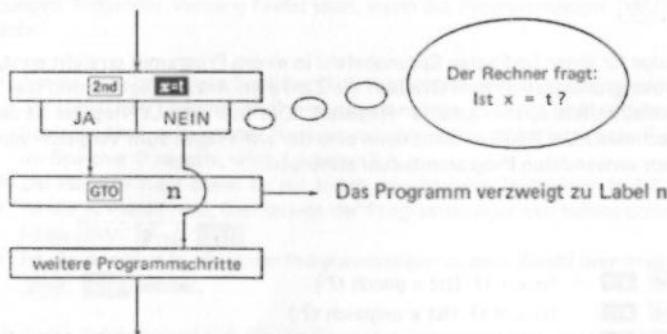
Wenn man diese Taste drückt, werden die Inhalte von Anzeigeregister und t -Register (Speicher 7) ausgetauscht. Damit haben Sie eine vorteilhafte Methode, einen Test- oder t -Wert in das Testregister zu bringen. (Sie können auch die Tastenfolge **2nd** **Exe** 7 oder einfach **STO** 7 verwenden, wenn dies für Ihr Programm geeignet ist.)

2nd **x=t** – Tastenfolge für die Frage: Ist x gleich t ?

Sie können diese Tastenfolge im Learn-Modus eingeben, um den Programmzeiger zu veranlassen, einen von mehreren Wegen durch das Programm zu wählen. Wenn der Programmzeiger beim Ablauf diesen Befehl erreicht, vergleicht der Rechner den Inhalt x des Anzeigeregisters mit dem Wert t im t -Register und fragt ab: Ist x gleich t ? Ist die Antwort ja, geht der Rechner zu dem Schritt weiter, der nach der Tastenfolge **2nd** **x=t** folgt. Bei negativer Antwort überspringt der Zeiger den Befehl nach **2nd** **x=t**, geht zum nächsten Befehl und setzt hier das Programm fort.

BEDINGTE SPRÜNGE (VERZWEIGUNGEN)

Nach der Tastenfolge **2nd** **x=!** kann ein beliebiger Befehl stehen, aber besonders zweckmäßig ist ein **GTO n** Befehl, der auch im folgenden Diagramm verwendet wird.



Hier noch einmal die wesentlichen Punkte, die bei Anwendung der Tastenfolge **2nd** **x=!** zu beachten sind:

- Der Rechner stellt die Frage: Ist $x = t$?
- Ist die Antwort ja — geht er zu dem unmittelbar nachfolgenden Schritt.
- Ist die Antwort nein — überspringt er den folgenden Schritt und fährt dann fort.

ANDERE BEDINGTE SPRÜNGE

INV **2nd** **x=t** — Tastenfolge für die Frage: Ist $x \neq t$?

Diese Tastenfolge wird im Learn-Modus eingegeben, um einen Programmsprung (oder eine Entscheidung) zu programmieren. Wenn der Programmzeiger die Tastenfolge für die Frage: Ist $x \neq t$? erreicht — **INV** **2nd** **x=t** — findet nachstehender Vorgang statt:

- Der Rechner vergleicht den Inhalt x des Anzeigeregisters mit dem Wert t im t-Register (Speicher 7).
- Der Rechner stellt dann die Frage: Ist $x \neq t$?
- Ist die Antwort positiv, rückt der Programmzeiger zu dem unmittelbar nachfolgenden Schritt weiter.
- Ist die Antwort negativ, überspringt der Programmzeiger den Folgebefehl und setzt das Programm fort.

Beachten Sie wiederum, daß nach der Tastenfolge **INV** **2nd** **x=t** ein beliebiger Programmschritt stehen kann, ein **GTO n** Befehl ist jedoch an diesen Punkten besonders geeignet.

[2nd] [x=] – Tastenfolge für die Frage: Ist x größer oder gleich t?

Wenn der Programmzeiger diesen Befehl im Programmablauf erreicht, wirkt er wie folgt:

- Der Rechner vergleicht die Werte x im Anzeigeregister und t im t-Register.
- Er stellt die Frage: Ist x größer oder gleich t?
- Bei positiver Antwort rückt er zu dem unmittelbar folgenden Befehl weiter.
- Bei negativer Antwort überspringt er den Folgebefehl und setzt das Programm fort.

[INV] [2nd] [x=] – Tastenfolge für die Frage: Ist x kleiner als t?

Wenn der Programmzeiger diesen Befehl im Programmablauf erreicht, wirkt er wie folgt:

- Der Rechner vergleicht die Werte x im Anzeigeregister und t im t-Register.
- Er stellt die Frage: Ist x kleiner als t?
- Ist die Antwort positiv, rückt er zu dem Schritt unmittelbar nach der Tastenfolge **[INV]** **[2nd] [x=]** weiter.
- Ist die Antwort negativ, überspringt er den Folgebefehl und setzt dann das Programm fort.

Ergänzende Anmerkungen

Es ist wirklich sehr einfach, sich diese Befehle zu merken, wenn man das Gesamtbild im Auge behält. Grundsätzlich können Sie den Rechner veranlassen, einen der vier Tests durchzuführen: Ist x gleich t, ist x größer oder gleich t, oder die Umkehrungen – ist x ungleich t, ist x kleiner als t? In jedem Fall testet der Rechner die Werte und stellt eine Frage. Wenn die Frage mit ja zu beantworten ist, rückt er zu dem unmittelbar folgenden Schritt weiter. Lautet die Antwort nein, überspringt er den folgenden Schritt und fährt fort.

Beachten Sie, daß die Größe x, die mit t verglichen wird, der Inhalt des Anzeigeregisters ist. Das Anzeigeregister hält die Zahlen mit 11 Stellen gespeichert, von denen jedoch nur 8 korrekt gerundete Stellen im Anzeigefeld ausgewiesen werden. In bestimmten Fällen können diese letzten 3 Stellen das Ergebnis der hier beschriebenen Vergleiche beeinflussen. (Diese Fälle kommen zwar selten vor, aber es ist eine Hilfe, über diese Möglichkeit informiert zu sein, wenn sich Probleme ergeben.)

Die Anwendung von Unterprogrammen ist eine vorteilhafte Technik, um Programmschritte zu sparen und ein umfangreiches Programm in leicht analysierbare Segmente zu zerlegen. Und schließlich gehören Unterprogramme zu den guten Programmierpraktiken. Ein Unterprogramm ist nichts anderes als ein „Miniprogramm“, das Sie als Teil eines Programms schreiben können. Immer, wenn Sie in einem Programm eine Gruppe von Schritten mehr als einmal anwenden müssen, empfiehlt sich ein Unterprogramm für diese Befehlgruppe.

Um eine Reihe von Programmschritten als Unterprogramm zu formulieren, geht man wie folgt vor:

- Die Reihe von Programmschritten wird mit einem Label begonnen.
- Die Reihe wird mit der Tastenfolge **[INV] [SBR]** abgeschlossen.

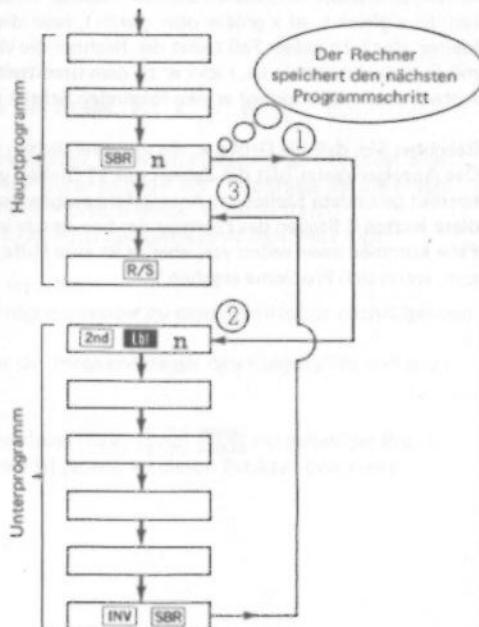
Die Länge eines Unterprogramms ist beliebig – das Gesamtprogramm mit allen seinen Unterprogrammen darf jedoch nicht die verfügbaren 50 Schritte überschreiten.

Anwendung eines Unterprogramms in einem Programm:

- Immer, wenn Sie ein Unterprogramm benötigen, drücken Sie im Learn-Modus **[SBR] n**, wobei n das Label des Unterprogramms ist. Sie können das Unterprogramm beliebig oft anwenden oder „aufrufen“.

Wenn der Programmzeiger die Tastenfolge **[SBR] n** im Programmablauf erreicht, führt der Rechner an Ihrer Stelle folgendes aus (siehe Diagramm):

1. Zuerst „notiert“ er den Programmschritt, der den **[SBR] n** Befehl enthält, in einem speziellen Speicher (dem Unterprogrammrücksprungregister).
2. Der Programmzeiger richtet sich auf Label n und beginnt mit der Durchführung der Befehle nach dem Label, bis der Befehl **[INV] [SBR]** erreicht wird.
3. Wenn der Programmzeiger den Befehl **[INV] [SBR]** erreicht hat, geht er an die Stelle zurück, wo das Unterprogramm aufgerufen wurde, und setzt das Programm fort.



Anwendung von Unterprogrammen direkt über die Tastatur:

Ein Vorteil von Unterprogrammen liegt darin, daß man sie auch separat vom Programm anwenden kann. Hierzu drücken Sie einfach **SBR n**, wenn der Learn-Modus aufgehoben ist. Der Programmzeiger stellt sich dann sofort auf Label **n** ein und führt das Unterprogramm durch. In diesem Fall wirkt die Tastenfolge **INV SBR** am Ende des Unterprogramms wie ein **R/S**-Befehl.

Anmerkungen zu Unterprogrammen:

Es gehört zu den guten Programmierpraktiken, Unterprogramme anzuwenden. Sie können damit Ihre Programme gut organisieren (und erleichtern auf diese Weise auch die Fehlersuche – sofern Fehler auftreten). Auch für andere Programmierer wird es leichter, Ihre Programme zu verstehen und anzuwenden, wenn sie in kleinere Unterprogramme gegliedert sind.

Ihr Rechner kann mit zwei Unterprogrammebenen arbeiten. Das heißt, daß während der Durchführung eines Unterprogramms ein weiteres Unterprogramm aufgerufen werden kann. Der Übergang zu einer dritten Ebene ist jedoch nicht möglich. Wenn man in der zweiten Ebene versucht, ein drittes Programm aufzurufen, wird die Fehlerbedingung durch Blinken der Anzeige ausgewiesen. Das Unterprogrammrücksprung-Register hat nur Kapazität für zwei Rücksprungadressen.

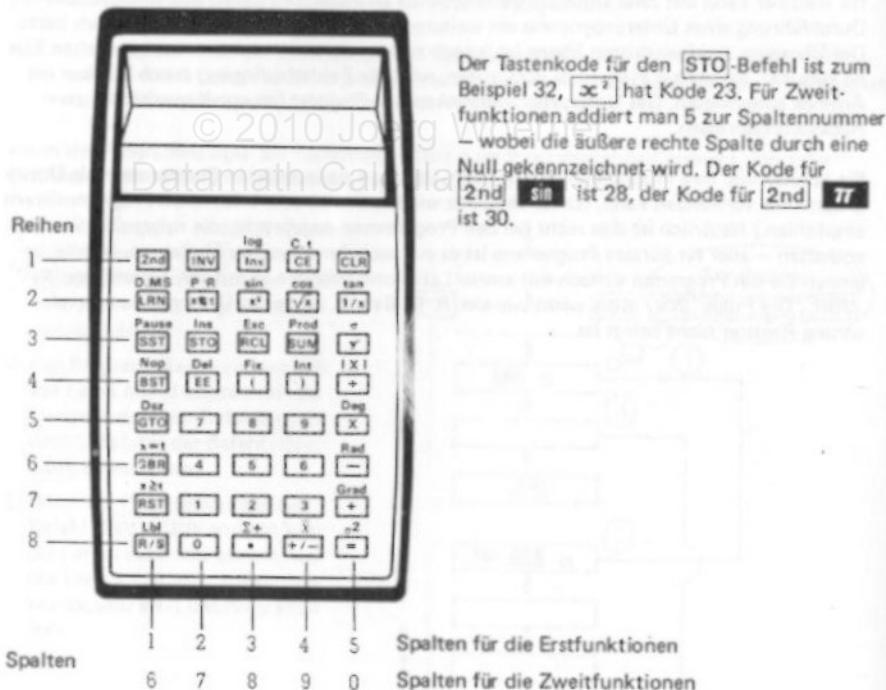
© 2010 Joerg Woerner

Sie können alle Ihre Programme so schreiben, daß man sie in anderen Programmen als Unterprogramme verwenden kann. (Diese Methode wird auch von sehr erfahrenen Programmierern empfohlen.) Natürlich ist dies nicht bei den Programmen angebracht, die nahezu 50 Schritte enthalten – aber für kürzere Programme ist es ein nachahmenswertes Verfahren. Hierfür beginnen Sie das Programm einfach mit einem Label und beenden es mit der Tastenfolge **INV SBR**. Die Folge **INV SBR** wirkt wie ein **R/S**-Befehl, wenn das Unterprogrammrücksprung-Register nicht belegt ist.

REDIGIEREN VON PROGRAMMEN TASTENKODES

Ihr Rechner verfügt über eine Reihe von Tasten, mit denen Sie Ihre Programme mühelos und schnell ändern können, um neue Problemlösungen festzulegen oder um eventuelle Fehler im Programm zu korrigieren. Mit diesen Redigiereigenschaften erreichen Sie jeden Punkt im Programm, Sie können ihn analysieren, einen Schritt ändern, Schritte einfügen oder entfernen, oder Leerstellen im Programm für spätere Verwendung freihalten.

Wie bereits erwähnt, speichert der Rechner die Programmschritte als eine Folge von Tastenkodes. Diese Kodes erhält man auf denkbar einfache Weise. Der Kode für eine Zifferntaste auf der Tastatur ist einfach die Ziffer selbst (00 bis 09). Der Tastenkode für eine Erstfunktion ist eine zweistellige Nummer: die erste Stelle gibt die Reihe an, in der die Taste liegt, die zweite Stelle leitet sich aus der Spalte her. (Die Reihen sind von 1 bis 8 von oben nach unten numeriert, die Spalten von 1 bis 5 von links nach rechts – siehe Abbildung).



Bei Eingabe eines Programms im Learn-Modus können Sie die Tastenkodes nicht beobachten, weil die Programmspeicher-Anzeige immer zum nächsten Schritt weiterrückt. Um die Kodes in einem Programm zu sehen, müssen Sie das Programm zunächst eingeben, dann zurückgehen und die Kodes überprüfen. Hierzu gehen Sie wie folgt vor:

REDIGIEREN VON PROGRAMMEN

SST

Ein einfaches Programm soll als Arbeitsgrundlage eingegeben werden. Es zeigt die Quadrate beim Abzählen an (1, 2, 3, etc.). Tasten Sie das Programm ein:

Tasten	Anzeige/Bemerkungen
AUS-EIN	0 Gesamtlösung
LRN	00 00
1	01 00
SUM	01 34 0
2	02 00
RCL	02 33 0
2	03 00
x²	04 00
2nd FUNC	05 00
RST	06 00
LRN RST	0

Für den Ablauf des Programms drücken Sie einfach **R/S**. Sie sehen in der Anzeige 1, 4, 9, 16, etc. Drücken Sie **R/S** und halten Sie die Taste kurz fest, um das Programm zu unterbrechen. Jetzt können Sie bei diesem Programm die Tastenkodes überprüfen und das Programm mit den Tasten redigieren, die auf dieser und auf der nächsten Seite beschrieben sind. (Drücken Sie **RST**, um an den Anfang des Programms zurückzugehen.)

SST – Einzelschritt

Verwendet man diese Taste im Learn-Modus, durchläuft sie Ihr Programm in Einzelschritten und ermöglicht die Überprüfung der Kodes. Beachten Sie, daß diese Taste keinen eigenen Kode hat und nicht als Programmschritt eingegeben werden kann. Die Anwendung dieser Taste ist auch möglich, wenn der Learn-Modus aufgehoben ist. In diesem Fall können Sie ein Programm in Einzelschritten durchführen. Für Redigierzwecke drücken Sie **SST** wiederholt, um die Kodes für Ihr Programm zu beobachten:

Tasten	Anzeige	Bemerkungen
LRN	00 01	1
SST	01 34 2	SUM 2
SST	02 33 2	RCL 2
SST	03 23	x²
SST	04 36	2nd FUNC
SST	05 71	Letzter Programm- schritt RST
SST	06 00	

REDIGIEREN VON PROGRAMMEN

BST

GTO 2nd nn

2nd Ins 2nd Ins

BST – Einzelschritt zurück

Diese Taste operiert nur im Learn-Modus. Sie können damit in Ihrem Programm in Einzelschritten zurückgehen. Drücken Sie **BST** wiederholt, um Ihre Programmcodes in Einzelschritten zurück zu überprüfen. (Diese Taste hat ebenfalls keinen eigenen Kode und kann nicht als Programmschritt eingegeben werden.)

GTO 2nd nn

Wie bereits erwähnt kann man mit dieser Tastenfolge direkt zu einem Programmschritt gehen, wenn der Rechner nicht in den Learn-Modus geschaltet ist. Drücken Sie **LRN**, um den Learn-Modus aufzuheben, und dann **GTO** 2nd 05, und **LRN** erneut. In der Anzeige erscheint 05 71 – Sie sind bei Schritt 5 des Programms, der den **RST**-Befehl enthält.

Änderung oder Redigieren eines Programms:

Man kann Programmschritte überschreiben, freihalten, einfügen oder löschen. Im folgenden ein kurzer Blick auf jede dieser Möglichkeiten:

Überschreiben eines Programmschritts:

Gehen Sie genau zu dem Schritt, den Sie überschreiben wollen, und geben Sie einfach den neuen Tastenbefehl oder die neue Tastenfolge ein. Der neue Befehl überschreibt den alten und ersetzt ihn. (Die Anzeige rückt zum nächsten Schritt weiter, wenn Sie diese Änderung durchführen. Sie können also zurückgehen und mit der Taste **BST** den neuen Kode prüfen.)

2nd Nop – Die Null-Operation (NO-OP)

Mit dieser Tastenfolge können Sie einen Befehl in Ihrem Programm mit einem NO-OP-Befehl aussparen. Er wird einfach vom Rechner ignoriert. Die Folge **2nd Nop** ermöglicht die Bildung freier Plätze in einem Programm, die man später belegen kann, oder die Herausnahme jedes gewünschten Befehls. Gehen Sie im Learn-Modus einfach zu dem Befehl, der durch einen Leerbefehl (NO-OP-Befehl) ersetzt werden soll, und drücken Sie **2nd Nop**. Damit kommt an Stelle des alten Befehls der Leerbefehl mit Kode 46.

2nd Ins – Einfügen (insert)

Mit dieser Folge kann man mühelos zusätzliche Schritte in das Programm einfügen. Wenn man Befehle einfügen will, geht man zu dem Programmschritt genau an die Stelle, wo zusätzliche Schritte aufgenommen werden sollen (der Rechner ist hierbei auf Learn geschaltet). Dann drückt man **2nd Ins**.

Alle Befehle von diesem Punkt an werden um einen Platz nach unten gerückt. An der Stelle des Programmspeichers, wo Sie sich befinden, bleibt ein freier Raum. Jetzt können Sie hier einen neuen Befehl eingeben. Wenn Sie mehr Befehle einfügen wollen, wiederholen Sie einfach diesen Vorgang.

Achtung: Jeder Schritt in Speicherplatz 49 geht verloren, wenn Sie **2nd Ins** drücken.

2nd Del – Löschen eines Einzelbefehls (delete)

Mit dieser Folge kann man einen Befehl aus einem Programm nehmen, und die entstandene Lücke schließen – dabei rücken alle folgenden Befehle um einen Speicherplatz auf. Gehen Sie einfach zu dem Befehl, der gelöscht werden soll, und drücken Sie **2nd Del**.

Wie schreibt man ein Programm?

Mit zunehmender Programmiererfahrung eignen Sie sich einen natürlichen Rhythmus an, und sobald Sie mit einem Problem konfrontiert werden, bietet sich die Methode, wie Sie die Lösung programmieren, von selbst an. Die Schritte, nach denen Sie tatsächlich vorgehen, sind natürlich von Ihrer individuellen Lösungsmethode abhängig. Aber hier sind einige Vorschläge:

1. Beschäftigen Sie sich mit der Aufgabe – Stellen Sie die Gleichungen und Verfahren zusammen, die Sie brauchen.
2. Setzen Sie sich selbst ein Ziel. Legen Sie fest, wie Sie das Programm anwenden. Dann wissen Sie, welche Werte einzugeben sind, welche Tasten gedrückt werden müssen, und was angezeigt werden soll, wenn das Programm abgeschlossen ist.
3. Planen Sie den Programmlauf – Bestimmen Sie, wie der Programmzeiger den Verarbeitungsfluß steuern soll (hier ist ein Flußdiagramm oder ein Ablaufplan für die Schritte eine große Hilfe).
4. Schreiben Sie die Programmschritte auf (oder geben Sie das Programm direkt in den Rechner ein, wenn Sie mit dem Gerät vertraut sind). Behalten Sie die ständige Kontrolle über die Werte, die die Speicher belegen, und über die Funktion der einzelnen Labels. Es gibt Punkte, die man leicht vergißt, und die dann häufig die Ursache für Fehler und Wiederholungen sind.
5. Überprüfen Sie das Programm nach der Eingabe mit bekannten Daten, um sicherzustellen, daß es richtig funktioniert.
6. Redigieren und korrigieren Sie bei Bedarf.
7. Schreiben Sie das Programm nieder und bewahren Sie es für spätere Anwendungen auf. Zu diesem Zweck werden Standardformulare mitgeliefert, die sich in ihrem Aufbau auf diese Anleitung beziehen. Sobald das Programm funktioniert, können Sie **RST LRN** drücken und dann mit der Taste **SST** die Tastenkodes durchgehen, und sie ebenfalls im Standardformular eintragen. Diese vollständige Aufzeichnung ermöglicht es Ihnen oder einem Bekannten, später das Formular zu nehmen, und das Programm zu verstehen – wie es funktioniert, wie man es anwendet, und wie man es testet.

ENTWICKLUNG DES PROGRAMMIERSTILS

Entwicklung des Programmierstils:

Ob Sie einen Rechner oder einen Großcomputer programmieren – Sie müssen immer bedenken, daß die Lösung jeder Aufgabe unterschiedlich programmiert werden kann. Mit zunehmender Programmiererfahrung werden auch Sie Ihren eigenen Stil entwickeln, die Kniffe kennenlernen, und bestimmte Techniken vorziehen. Wenn Sie mehr Übung bekommen, sollten Sie von Zeit zu Zeit Ihre Programmietechniken daraufhin prüfen, ob Sie die Leistungsfähigkeit des Geräts voll nutzen. Scheuen Sie sich nicht, neue Wege zu gehen, neue Alternativen zu probieren und mit neuen Methoden zu experimentieren. Programmieren ist eine ausgezeichnete Übung für klares, logisches Denken, und viele finden daran Vergnügen (und manche sind dieser Neigung ganz ergeben). Ihr Rechner ist mit Bedacht so angelegt, daß die Anfänge einfach sind, und daß Sie dann sehr schnell Experte im Programmieren werden.

EIN AUSBLICK:

Ein weiterer Punkt – Ihr Rechner ist einfach in der Anwendung, sofort funktionsbereit und finanziell erschwinglich. Dennoch vereinigt er in sich alle Hauptmerkmale und Funktionen eines Großcomputers. Das Konzept eines Programmspeichers, ein Programmzeiger, der sich durch den Speicher bewegt, Labels, Sprungbefehle, Schleifen und bedingte Verzweigungen haben alle programmierbaren Digitalrechner gemeinsam. Wenn Sie einmal Ihren Rechner programmieren können, ist es für Sie einfach, darauf aufzubauen und mit der Programmierung anderer Geräte – unabhängig von deren Leistung – fortzufahren. Größere Computer akzeptieren nur ihre Eingangsinformationen in anderer Form, haben mehr Möglichkeiten, Informationen auszugeben und sind im allgemeinen so ausgelegt, daß sie mehr Informationen verarbeiten und speichern können. Häufig lernt der Anfänger diese Geräte dadurch kennen, daß er die Befehle nacheinander auf Karten, Papierstreifen oder Magnetplatten schreibt. Diese Einzelbefehle gleichen Ihren Tastenbefehlen – besonders, wenn man beginnt, Computersprachen wie FORTRAN oder BASIC zu lernen. Mit dem Aufbau der Tastatur Ihres Rechners können Sie mühelos die Programmierung in einer dieser Sprachen lernen, sobald Sie die Grundlagen Ihres Rechners verstanden haben.

Im nächsten Jahrzehnt wird die Datenverarbeitung zunehmend das tägliche Leben bestimmen. Programmierbare Geräte werden dann überall eingesetzt – in der Küche, in der Kontrolle der privaten finanziellen Verhältnisse, in der Kommunikation, für persönliche Sicherheit. Wenn Sie heute die Grundlagen der Programmierung mit Ihrem Rechner lernen, sind Sie auf den zunehmenden Einsatz dieser Geräte gut vorbereitet – Sie können Freude daran haben und sie als einen selbstverständlichen Teil Ihres Alltags sehen.

Trotz größter Vorsicht macht man hin und wieder Fehler. Die Anwendung Ihres Rechners ist zwar sehr einfach und logisch, aber manchmal überseht man einen Programmierfehler, und das Programm funktioniert nicht einwandfrei, auch wenn es sorgfältig eingetastet wurde. Hier sind einige Tips und Punkte, die Sie bei der Fehlersuche in Programmen beachten sollten.

Ihr Rechner zeigt Fehler in einem Programm auf verschiedene Weise an. Wenn an einer Stelle im Programm eine falsche Tasten- oder Operationsfolge zu einer Fehlerbedingung führt (siehe Anhang C), wird das Programm unterbrochen und die Anzeige blinkt. Das heißt, Sie haben in einem Programm eine Operation verlangt, die auch über die Tastatur nicht möglich ist.

Wenn Sie die Grenzen der Anzeige überschreiten oder eine falsche Tastenfolge anwenden, wird die Verarbeitung unterbrochen. In anderen Fällen wird die Fehlerbedingung nicht durch die blinkende Anzeige angekündigt, dennoch sind die Ergebnisse Ihres Programms nicht richtig. Es empfiehlt sich, die Genauigkeit eines Programms mit bekannten Testdaten und -ergebnissen zu sichern. Wenn Sie nicht das erwartete Resultat erhalten, muß die Ursache des Problems gesucht werden.

Wenn ein Programm nicht funktioniert – versuchen Sie, „wie der Rechner zu denken“. Gehen Sie noch einmal Ihren Ablaufplan durch (und Ihre Tastenkodes) und führen Sie genau das aus, was hier angewiesen ist. Sie stellen vielleicht fest, daß Sie das Programm versehentlich in eine Sackgasse gebracht haben. Die Taste **SST** kann hier eine Hilfe sein, im Learn-Modus und im Rechen-Modus.

RST – Achten Sie darauf, ob Sie den Rechner überall rückgestellt haben, wo es nötig ist – um zum Programmanfang zu kommen. Außerdem muß vor Beginn eines neuen Programms alles gelöscht werden. Schalten Sie also grundsätzlich den Rechner kurz aus und wieder ein.

– Algebraisches Operationssystem –

Bedenken Sie, daß die gesamte Leistung des AOSTM-Eingabesystems auch in den Programmen für Sie arbeitet. Alle Operationen werden exakt nach der mathematischen Hierarchie ausgeführt. Diese Reihenfolge muß nicht der Eingabefolge entsprechen, die Sie in Ihrem Programm verwendet haben.

Berücksichtigen Sie, daß Ihr Rechner gleichzeitig bis zu 4 unvollständige Operationen aufnehmen kann.

Wenn Sie Zweifel haben, in welcher Reihenfolge die Ausdrücke berechnet werden, denken Sie daran, daß Sie Klammern setzen können: **()** – damit können Sie sicherstellen, daß die Berechnungen in der von Ihnen angegebenen Reihenfolge stattfinden.

FEHLERSUCHE

Beachten Sie, daß bei den Funktionen mit einer Variablen die entsprechende Funktionstaste nach der Zahl folgen muß, mit der gerechnet wird. Wenn Sie in einem Programm den Sinus von 30° ausrechnen wollen, ist die Tastenfolge 30 **2nd sin**.

= Taste — Sie schließt alle unvollständigen Operationen ab, die Rechner-intern noch offen sind — Vorsicht also bei der Anwendung. (Besondere Sorgfalt ist bei der Anwendung von **=** in Unterprogrammen angebracht. Hier setzen Sie besser Klammern, **()**, um die Berechnungen in einem Unterprogramm abzuschließen, weil mit **=** auch im Hauptprogramm alle Berechnungen abgeschlossen werden.)

Labels — Jedes der 10 Labels darf nur ein einziges Mal in einem Programm vorkommen.

Winkelmodus — Beim Rechnen mit trigonometrischen Funktionen müssen Sie bedenken, daß nach dem Einschalten alle Winkel in der Einheit Altkl. ausgedrückt werden, wenn Sie die Winkeleinheit nicht mit einer der Tastenfolgen **2nd Rad** oder **2nd Grad** ändern. Die Winkeleinheit wird immer so lange beibehalten, bis Sie selbst eine andere wählen, (oder den Rechner aus- und einschalten).

Speicher — Prüfen Sie sorgfältig, ob bei allen Operationen mit Speichern auch der richtige Speicher verwendet wird. Wenn in Ihrem Programm vier Operationen in einem Ausdruck unvollständig bleiben, müssen Sie beachten, daß die Speicher 5 und 6 intern für die Speicherung von Teilen dieser Ausdrücke belegt werden. Alle zuvor in den Speichern 5 und 6 gespeicherten Zahlen gehen verloren, wenn Sie später einen langen Ausdruck mit vier unvollständigen Operationen eingeben. (Dazu kommt, daß Speicher 7 als t-Register eine besondere Funktion hat.) Speicher Null wird in Verbindung mit der Tastenfolge **2nd Osz** gebraucht — vergessen Sie auch diese Tatsache nicht.

Sie werden feststellen, daß Ihre Programme mit ein wenig Erfahrung korrekt und störungsfrei ablaufen. Sie werden auch einen Sinn für potentielle Schwierigkeiten entwickeln, sobald Sie einmal ein Problem hatten — die beste Art, Fehlersuche zu erlernen, ist, den Rechner ständig zu benutzen. Nach kurzer Zeit werden Sie nur noch selten gezwungen sein, Fehler in Ihren Programmen herauszufinden.

- IM HANDEL !

- ## • ABZÄHLEN.

- FAKULTÄT

- #### ■ KOMBINATIONEN

- ## ■ INTEGRALE RECHNUNG

- ## • LINEARE REGRESSION

- #### • HI-LO-ZAHLENRATESPIEL

Angenommen, Sie arbeiten in einem Warenhaus. Der Leiter des Hauses hält Preissenkungen für angebracht. Tatsächlich weist er seine Angestellten oft an, die einzelnen Artikel niedriger auszuzeichnen. Oft variiert der Preisnachlaß von einer Abteilung zur anderen, 25% bei Schuhen, 15% bei Eisenwaren, etc. In diesem Programmbeispiel werden Kenntnisse aufgefrischt und die Anwendung der Speicher als Teile von Programmen wird ausgedehnt.



Ziel

Sie wollen, daß Sie in jede Ihnen zugewiesene Abteilung des Warenhauses gehen, den Preis eines Artikels im Regal eintasten, **R/S** drücken, und dann den herabgesetzten Preis erhalten können. Sie wollen auch in der Lage sein, den Preisnachlaß so einfach wie Ihr Manager von einer Abteilung zur anderen zu variieren.



Planung des Programms

Wenn der Preis eines Artikels um 25% sinkt, können Sie den neuen Preis mit folgender Formel berechnen: Neuer Preis = alter Preis X (1 – .25). Tatsächlich kann bei jedem Preisnachlaß der neue Preis mit derselben Formel ermittelt werden:

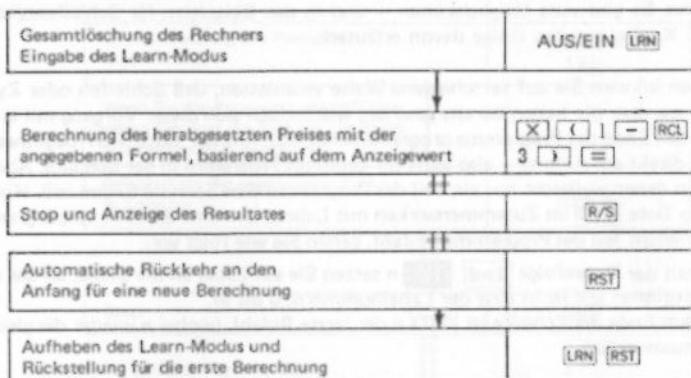
$$\text{Neuer Preis} = \text{alter Preis} \times (1 - \text{Rabatt})$$

In diesem Programm wird der Rabatt als Dezimalzahl (25% = 0.25) in Speicher 3 gespeichert. Wenn man diesen Wert dann zur Berechnung des neuen Preises benötigt, wird er aufgerufen.

$$\text{Neuer Preis} = \text{alter Preis} \times (1 - \boxed{\text{RCL}} 3)$$

Der Vorteil dieser Methode in Ihrem Programm: Wenn Sie in eine andere Abteilung mit einem anderen Rabatt gehen, müssen Sie den neuen Rabatt nur in 3 speichern, dann können Sie fortfahren. Diese Technik ist in vielen Situationen angebracht, in denen Sie eine Zahl in einem Programm ändern wollen, ohne das gesamte Programm neu zu schreiben.

Der Programmablaufplan



Geben Sie Ihr Programm sorgfältig ein, genau nach den Tastenbefehlen im Ablaufplan.

Ablauf des Programms

© 2010 Joerg Woerner

Der Warenhausleiter beschloß, die Schuhe zum Preis von 14.95, 17.99 und 18.99 um 25 % billiger zu verkaufen. Der Preis für die Bücher, die jetzt 6.00, 7.99 und 17,50 kosten, soll um 13 % gesenkt werden. Wie hoch sind die neuen Preise?

Tasten	Anzeige	Bemerkungen
$\text{[2nd]} \text{ [L1] 2}$	0.00	Anweisung, den Anzeigewert auf 2 Dezimalstellen zu runden (leichtes Ablesen von Währungseinheiten)
0.25 [STO] 3	0.25	Der erste Rabatt wird als Dezimalzahl in Speicher 3 gespeichert

Man gibt die alten Preise ein, drückt [R/S] und liest die neuen Preise ab:

14.95 [R/S]	11.21
17.99 [R/S]	13.49
18.99 [R/S]	14.24
0.13 [STO] 3	0.13

Speicherung des neuen Rabatts

Eingabe der Preise, die um 13 % reduziert werden, Taste [R/S] und Ablesen der neuen Preise:

6.00 [R/S]	5.22
7.99 [R/S]	6.95
17.50 [R/S]	15.23

Beachten Sie, daß man durch Speicherung des Preisnachlasses im Speicher 3 die einzelnen Nachlässe der verschiedenen Abteilungen leicht ändern kann.

Nächste Aufgabe

Sie wollen die Kontrolle über alle Preisbewegungen eines Tages haben. Können Sie das obige Programm so ändern, daß Sie über den Speicher 4 die Kontrolle über die Anzahl der im Preis reduzierten Artikel enthalten? (Siehe Lösungsvorschlag A-10)

GTO **n** **UND** **2nd** **Lbl** **n**

Die Erfahrung hat gezeigt, daß es vorteilhaft ist, wenn man innerhalb der Programme zählen kann. Es gibt viele Möglichkeiten – und in den Beispielen für Schleifenbildung in diesem Kapitel werden einige davon erläutert.

Daneben können Sie auf verschiedene Weise veranlassen, daß Schleifen oder Zyklen durchlaufen werden. Sie haben bereits gesehen, wie einfach sich dieser Vorgang mit einem **RST** Befehl am Ende des Programms programmieren läßt. Mit der Taste **RST** beginnen Sie jedoch immer direkt am Anfang – also läuft Ihr gesamtes Programm in der Schleife. Aber es gibt Fälle, in denen vielleicht nur ein Teil des Programms eine Schleife bilden soll. Hierfür eignet sich die Taste **GTO** im Zusammenwirken mit Labels. Um eine Schleife zu programmieren, die nur einen Teil des Programms umfaßt, gehen Sie wie folgt vor:

- Mit der Tastenfolge **2nd** **Lbl** **n** setzen Sie ein Label an die Stelle, an der die Schleife beginnen soll (**n** ist eine der Labelnummern, 0 bis 9).
- Am Ende der Schleife ist **GTO** **n** der letzte Befehl, (wobei **n** wieder die gleiche Labelnummer ist).

Hier ein Zähleispiel mit Schleifen:



Ziel

© 2010 Joerg Woerner

Datamath Calculator Museum

Sie wollen ein Programm schreiben, mit dem der Rechner in Abständen von 1, 2, 4, etc. – welche Zahl Sie auch wünschen – zählen kann. Sie möchten die Zählung von jeder beliebigen gewählten Zahl aus beginnen können. Das Programm soll wie folgt arbeiten:

- Sie geben die Zahl ein, in deren Abständen Sie zählen wollen und drücken **R/S**.
 - Sie geben eine Anfangszahl ein und drücken **R/S**
- Der Rechner zählt dann, beginnend mit Ihrer Anfangszahl, im angegebenen Abstand weiter.



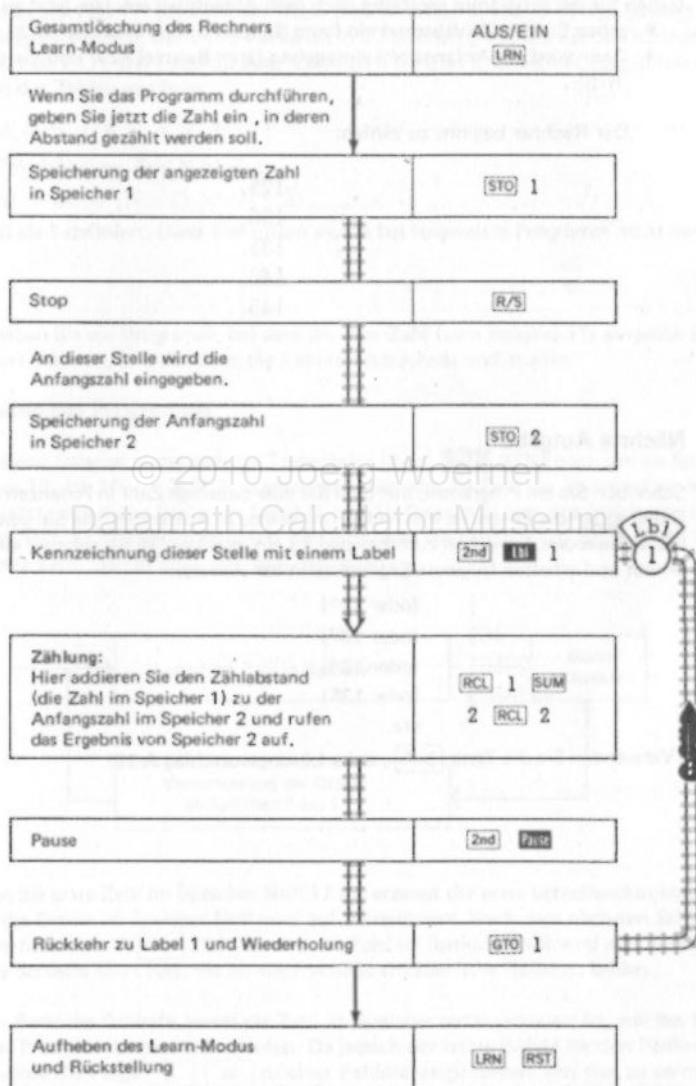
Planung des Programms

Im ersten Teil des Programms wird die erste Eingabe – der Abstand, in dem gezählt werden soll – in Speicher 1 eingebracht. Dann wird die Anfangszahl für die Zählung in Speicher 2 gespeichert. Nun, nach Speicherung dieser beiden Zahlen wollen Sie für die Zählfolge nicht zu diesem Programmteil zurückgehen, also setzen Sie ein Label – und weisen damit den Rechner an, daß dies der Rückkehrpunkt in der Schleife ist.

Nach dem Label schreiben Sie den Zähleil des Programms. Eine Möglichkeit ist der Aufruf der Zahl, in deren Abstand weitergezählt wird (Speicher 1). Diese Zahl wird dann zur Anfangszahl (Speicher 2) addiert. Dies ist die erste Zahl der Zählfolge. Deshalb rufen Sie den Inhalt von Speicher 2 auf, und geben zum Ablesen eine Pausen-Anweisung. Dann wiederholen Sie den Vorgang durch Rückkehr zum Label.

ABZÄHLEN: SCHLEIFEN MIT GTO n UND 2nd Lbl n

Der Programmablaufplan



ABZÄHLEN: SCHLEIFEN MIT

GTO n UND 2nd n



Ablauf des Programms

Geben Sie das Programm sorgfältig nach dem Ablaufplan ein. Um jetzt zu zählen,

- geben Sie den Zählabstand ein (zum Beispiel 5) und drücken R/S
- Dann wird die Anfangszahl eingegeben (zum Beispiel 120) und Sie drücken wieder R/S .

Der Rechner beginnt zu zählen:

125.

130.

135.

140.

145.



Nächste Aufgabe

© 2010 Joerg Woerner

Schreiben Sie ein Programm, mit dem Sie eine beliebige Zahl in Potenzen erheben können. Dabei sollen die Potenzen von Null an aufwärts, in Abständen, die Sie gewählt haben, berechnet werden. Sie geben zum Beispiel 13 ein, drücken R/S, geben 2 ein, drücken R/S erneut und erhalten folgende Ergebnisse in der Anzeige:

1. (oder 13^0)

169. (oder 13^2)

28561. (oder 13^4)

4826809. (oder 13^6)

etc.

(Verwenden Sie die Taste **y^x**, siehe Lösungsvorschlag A-10)

Oft findet man in Wahrscheinlichkeitsberechnungen und in der Statistik den Begriff der Fakultät einer Zahl. Die Fakultät einer Zahl wird durch ein Ausrufezeichen symbolisiert. 17! Fakultät wird als 17! geschrieben. Wenn Sie sich schon einmal die Frage gestellt haben, auf wieviel verschiedene Weisen 17 Personen in 17 Stühlen gruppiert werden können, dann ist die Antwort 17!. Die Fakultät einer Zahl, zum Beispiel 5, errechnet man durch Ausmultiplizieren der Zahlen wie folgt:

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

Die Fakultät einer Zahl N ist also:

$$N! = N \times (N-1) \times (N-2) \dots \dots \times 1.$$

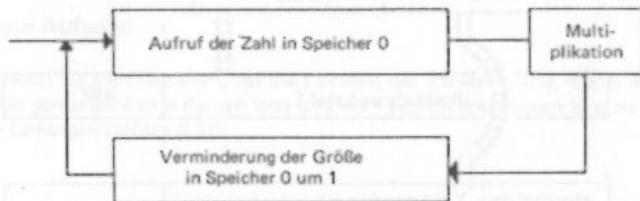
(0! ist als 1 definiert. Diese Definition wurde bei folgendem Programm nicht berücksichtigt.)

Ziel

Schreiben Sie ein Programm, bei dem Sie eine Zahl (zum Beispiel 17) eingeben und R/S drücken können, und das dann die Fakultät berechnet und anzeigt.

Planung des Programms

Für dieses Beispiel bietet sich die Tastenfolge 2nd Dsz GTO n an, um im Speicher 0 die Zahlen 17, 16, 15, ..., 3, 2, 1 zu erzeugen. Man gibt die Zahl ein, speichert sie im Speicher 0, und setzt nach dieser Stelle ein Label (Label 1). Dann ruft man den Inhalt von Speicher 0 auf, drückt die Multiplikationstaste und gibt dann die Folge 2nd Dsz GTO 1 ein. Die Schleife kann man in etwa wie folgt darstellen:

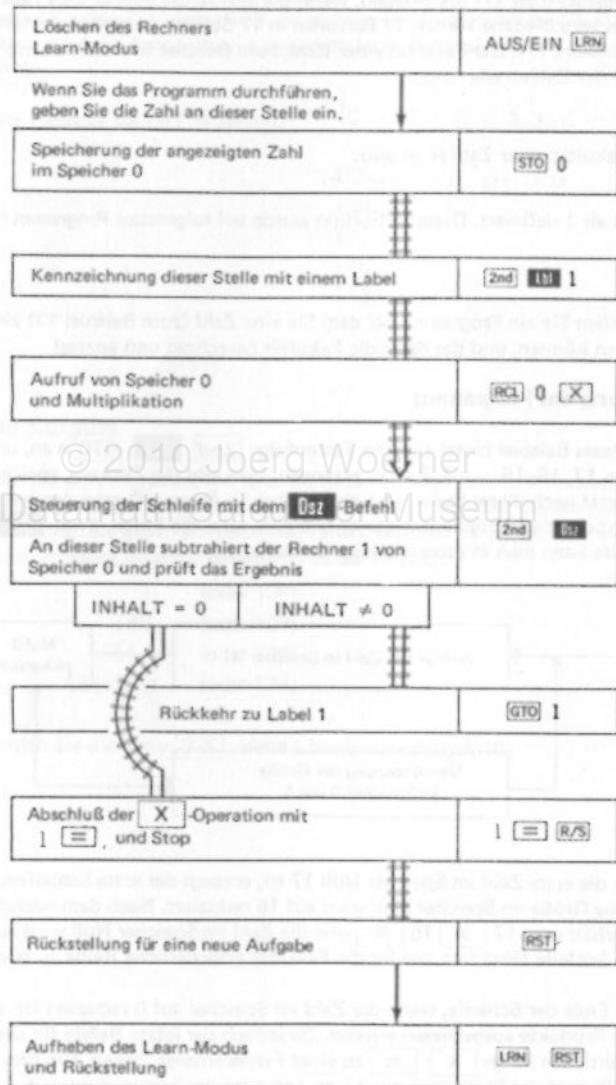


Wenn die erste Zahl im Speicher Null 17 ist, erzeugt der erste Schleifendurchlauf 17 X, und die Größe im Speicher Null wird auf 16 reduziert. Nach dem nächsten Schleifendurchlauf erhält man 17 X 16 X und die Zahl im Speicher Null wird auf 15 vermindert. Diese Schleife fährt fort, die für die Fakultät erforderliche Reihe zu bilden.

Nach Ende der Schleife, wenn die Zahl im Speicher auf 0 reduziert ist, soll das Ergebnis dieser Produkte ausgewiesen werden. Da jedoch der letzte Befehl für den Rechner ein X ist, würde die Folge X = zu einer Fehleranzeige führen. Um dies zu vermeiden, schließt man das Programm mit 1 = ab, und das Endprodukt, X 1 = gibt das richtige Ergebnis an.



Programmablaufplan



FAKULTÄT! SCHLEIFEN MIT

2nd **D** GTO n

VERDANKEN

Ablauf des Programms

Geben Sie das Programm sorgfältig nach dem Ablaufplan ein. Wenn die Eingabe abgeschlossen ist, kann man die Fakultät jeder (positiven ganzen Zahl kleiner als 70) berechnen: Man gibt einfach die Zahl ein und drückt **R/S**.

Einige Beispiele:

Tasten	Anzeige/Bemerkungen
5 R/S	120. oder 5!
10 R/S	3628800 oder 10!
17 R/S	3.5568743 14 oder 17!
50 R/S	3.0414093 64 oder 50!

Beachten Sie, daß Ihr Rechner für diese letzte Berechnung einige Zeit benötigt. Denken Sie daran, daß jedesmal, wenn Sie eine Schleife programmieren, der Rechner genügend Zeit zur Ermittlung des Resultats braucht. Einige Schleifen können Stunden – ja sogar Tage dauern! Beachten Sie auch, daß das Programm für 0! ein falsches Ergebnis liefert. Wenn Sie 0 eingeben und die Taste **R/S** drücken, ist das Ergebnis nicht 1, sondern 0. Vielleicht wollen Sie ein Programm entwickeln, das auch 0! korrekt berechnet.

Wie bereits erwähnt, ist die Tastenfolge **2nd D** für eine Vielzahl von Situationen geeignet. In diesem Abschnitt wurden einige der direkten Anwendungsmöglichkeiten aufgezeigt.

Nächste Aufgabe

Entwickeln Sie ein Programm, das die Fakultät der Zahlen 1, 2, 3, 4, etc. anzeigt. Sie wollen, daß jede ganze Zahl zwei Pausen lang und die Fakultät drei Pausen lang ausgewiesen wird. (Siehe Lösungsvorschlag A-10)

KOMBINATIONEN

SBR UND INV SBR

Hierzu ein Programmbeispiel aus der Statistik, in dem auch Unterprogramme verwendet werden. Die Berechnung der Kombinationen ist dann erforderlich, wenn Objekte in Gruppen angeordnet werden (bei 10 Objekten werden zum Beispiel 3 gleichzeitig ohne Wiederholungen entnommen). Die allgemeine Formel für die Anzahl der Möglichkeiten, n Objekte zu kombinieren, wird als $\binom{n}{r}$ oder als $n C_r$ geschrieben und lautet: $\frac{n!}{(n-r)!r!}$

(Denken Sie daran, daß $n!$ = n Fakultät.)



Ziel

Das Programm soll so geschrieben sein, daß Sie die Gesamtzahl der Objekte (n) eingeben und dann R/S drücken können. Dann wollen Sie die Anzahl der Objekte in einer Gruppe (r) eingeben, die Taste R/S drücken, und der Rechner soll die Anzahl der Kombinationen für n Objekte anzeigen, wenn r Objekte gleichzeitig entnommen werden. ($n C_r$)



Planung des Programms

Zur Berechnung der Fakultät verwendet man ein Unterprogramm, das mit dem 2nd Befehl arbeitet. Hierzu kann man das bereits beschriebene Fakultätsprogramm heranziehen und es zum Unterprogramm umformen: Man beginnt die Folge mit einem Label (Label 2) und beendet sie mit der Tastenfolge INV SBR.

Man berechnet $n!$ und speichert den Wert in Speicher 1. Dann berechnet man $r!$ und speichert das Ergebnis im Speicher 2. Schließlich ermittelt man noch $(n-r)!$ und speichert diesen Wert in 3; die Resultate verwenden man, um

$$n C_r = \frac{n!}{(n-r)! r!}$$

zu erhalten.

Geben Sie das Programm sorgfältig nach dem Ablaufplan auf der nächsten Seite ein.



Ablauf des Programms

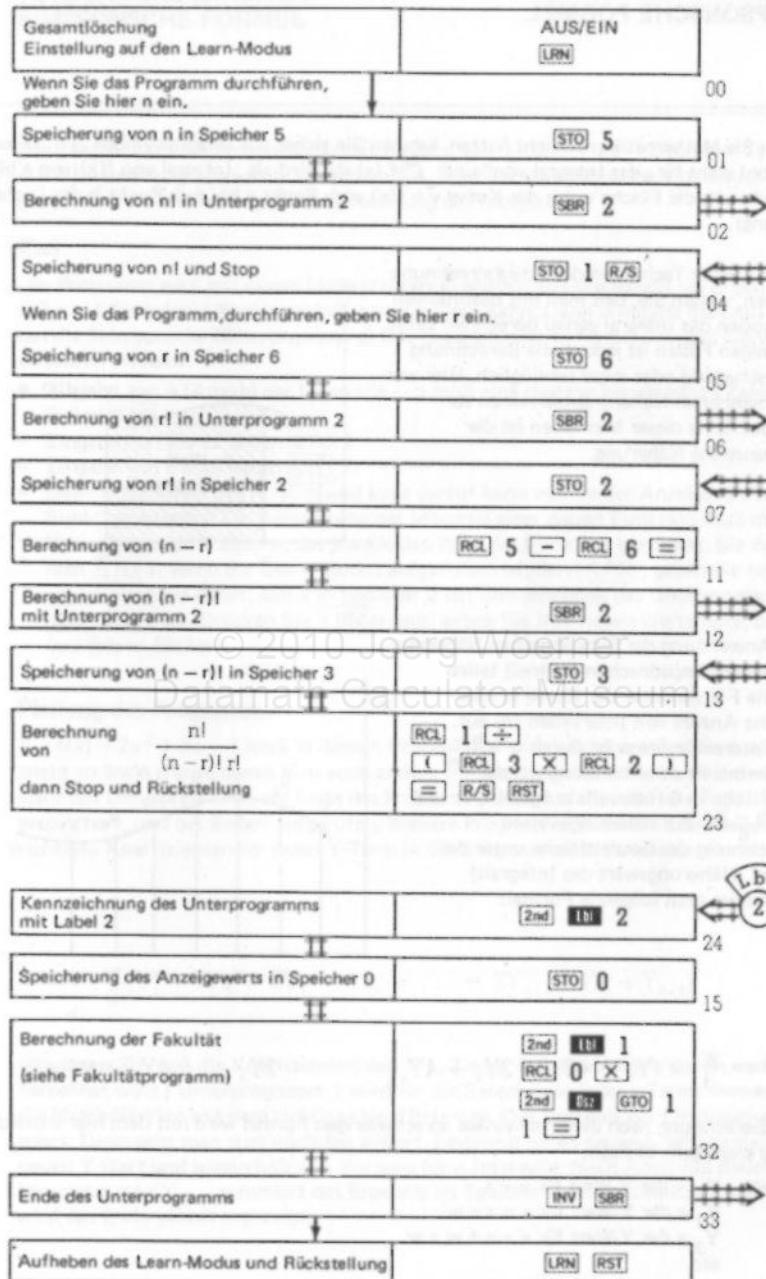
Berechnen Sie mit Hilfe Ihres Programms die mögliche Anzahl der Einzelspiele (5 Spielkarten), die aus einem Blatt von 52 Karten gegeben werden können. (${}_{52}C_5$)

Tasten	Anzeige/Bemerkungen
52 R/S	8.0658175 67 (das entspricht 52!)
5 R/S	Achtung! Lassen Sie genügend Zeit für den Abschluß dieser Berechnung 2598960. die Anzahl der möglichen Einzelspiele aus einem Blatt von 52 Karten (${}_{52}C_5$)

Jetzt soll nur das Unterprogramm verwendet werden, um 10! zu berechnen.

10 SBR 2 3628800. (das ist 10!)

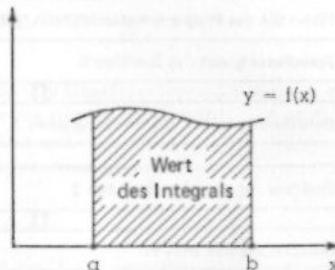
Programmablaufplan



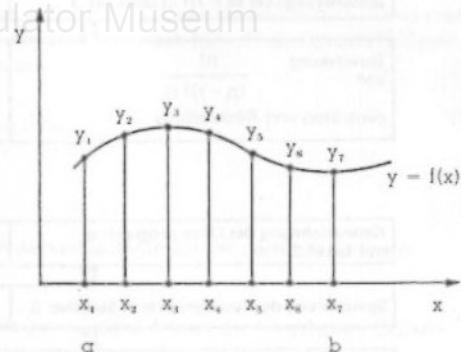
INTEGRALRECHNUNG: SIMPSONSCHE FORMEL

Wenn Sie Mathematikunterricht hatten, kennen Sie sicher das Integralsymbol „ \int “ . Dieses Symbol steht für „das Integral von“ und $\int_a^b f(x) dx$ wird als „Integral von $f(x)$ von a bis b “ – und stellt die Fläche unter der Kurve $y = f(x)$ vom Punkt a bis zum Punkt b dar (siehe Abbildung).

Wenn Sie die Techniken der Integralrechnung kennen, wissen Sie, daß man mit bestimmten Methoden das Integral genau berechnen kann. In einigen Fällen ist jedoch die Berechnung sehr schwierig oder sogar unmöglich. Hier werden manchmal Näherungstechniken verwendet. Eine dieser Methoden ist die Simpsonsche Näherung.



Bei Anwendung der Simpsonschen Näherung (oder der Simpsonschen Formel) teilen Sie die Fläche unter der Kurve in eine gleiche Anzahl von Intervallen (n) auf. Die Intervallbreite w ist durch $w = \frac{b-a}{n}$ bestimmt. In der Abbildung wurde die Fläche in 6 Intervalle aufgeteilt, so daß $w = \frac{b-a}{6}$. Zur näherungsweisen Berechnung der Gesamtfläche unter der Kurve (Näherungswert des Integrals) verwendet man folgende Formel:



$$\text{Fläche} = \frac{w}{3} \times (Y_1 + 4Y_2 + 2Y_3 + 4Y_4 + \dots + 2Y_{n-1} + 4Y_n + Y_{n+1}).$$

Die Berechnung nach dieser scheinbar so schwierigen Formel wird mit dem hier entwickelten Programm ganz einfach.

- Es gilt:
- Y_1 = der Y-Wert für $x = a$
 - Y_2 = der Y-Wert für $x = a + w$
 - Y_3 = der Y-Wert für $x = a + w + w$
 - etc.
 - Y_{n+1} = der Y-Wert für $x = b$

INTEGRALRECHNUNG : SIMPSONSCHE FORMEL

Als Beispiel wird ein Programm für das Integral von $y = 2x^2 + 3x - 1$ von $a = 1$ bis $b = 6$ mit $n = 6$ Intervallen geschrieben. (Beachten Sie, je größer die Anzahl der Intervalle, desto genauer wird die näherungsweise Berechnung des Integrals.)

Ziel

Das Programm wird mit einem Unterprogramm erstellt, das wiederholt die Gleichung $x = 2x^2 + 3x - 1$ für jeden x -Wert zwischen a und b lösen kann. Dann ermittelt man mit diesen Werten die Simpsonsche Näherungslösung. Das Programm ist so formuliert, daß Sie wie folgt verfahren müssen:

- Eingabe von n (Anzahl der Intervalle — n muß eine positive gerade ganze Zahl sein) und Tasten **STO 0**.
- Eingabe von b und Tasten **STO 1**.
- Eingabe von a und Tasten **STO 2**.
- Dann drückt man **RST** **R/S** und kurz darauf kann man in der Anzeige das Integral der Funktion ablesen. Zur Berechnung des Integrals einer neuen Funktion muß man nur das Unterprogramm 1 ändern, das jeweils den Wert der Funktion bestimmt. Sie drücken einfach **GTO 1**, wenn der Learn-Modus aufgehoben ist, dann **LRN**, geben die neue Funktion ein (denken Sie daran, daß x in Speicher 2 ist) und schließen das Unterprogramm mit **INV** **SBR** ab. Drücken Sie **LRN** erneut, geben Sie Ihre neuen Werte für a , b und n ein, und fahren Sie fort.

Planung des Programms

$Y = f(x) = 2x^2 + 3x - 1$ muß in diesem Programm oft berechnet werden, und außerdem leicht zu ändern sein, damit man auch andere Integrale berechnen kann. Aus diesem Grund steht das Unterprogramm am Ende des Programms. Am Anfang wird $w = \frac{b-a}{n}$ ermittelt, gespeichert, und bei Bedarf aufgerufen. Weitere Programmschritte sind einbezogen, um die richtigen Koeffizienten für jeden Y -Term in der Simpsomschen Formel zu finden:

$$\text{Fläche} = \frac{w}{3} (Y_1 + 4Y_2 + 2Y_3 + 4Y_4 + \dots + 2Y_{n-1} + 4Y_n + Y_{n+1}).$$

(Beachten Sie, daß die Koeffizienten von Y_1 , 2 oder 4 sind, abhängig davon, welcher Term berechnet wird.) Unterprogramm 1 wird für die Berechnung jedes y -Terms verwendet und für die Multiplikation mit dem richtigen Koeffizienten. Das Ergebnis wird in Speicher 3 summiert. Dann geht man zum nächsten x -Wert, bestimmt einen neuen Koeffizienten, einen neuen Y -Wert und wiederholt den Vorgang für n Intervalle. Nach Abschluß dieser Schritte berechnet man Y_{n+1} , summiert das Ergebnis im Speicher 3 und multipliziert mit $\frac{w}{3}$. Dann wird das Endergebnis angezeigt.

INTEGRALRECHNUNG: SIMPSONSCHE FORMEL

Geben Sie das Programm sorgfältig nach dem Ablaufplan auf der folgenden Seite ein.

Ablauf des Programms



Denken Sie daran, daß das Programm das Integral der Funktion

$y = 2x^2 + 3x - 1$ berechnet. Um das Programm auf eine andere Funktion umzustellen:

- Heben Sie den Learn-Modus auf
- Drücken Sie **GTO 1**
- Drücken Sie **LRN**
- Geben Sie die neue Funktion ein.
- Beenden Sie die Folge mit **INV SBR**

Berechnung von $2x^2 + 3x - 1$ für $a = 1$, $b = 5$, $n = 6$

Tasten

6 **STO** 0
5 **STO** 1
1 **STO** 2

RST R/S

Anzeige/Bemerkungen

6. Eingabe von n
5. Eingabe von b
1. Eingabe von a

114.66667

Der Näherungswert für das Integral
— der Rechner braucht etwas Zeit
für diese Berechnung

INTEGRALRECHNUNG: SIMPSONSCHE FORMEL

BRUNNEN
VÖLKL

Programmablaufplan

Gesamtlösung – Learn-Modus

AUS/EIN LRN

Berechnung von $\frac{b-a}{n}$

und Speicherung in 1 und 5

RCL 2	INV	SUM 1
RCL 0	INV	2nd F 1
RCL 1	STO 5 0	STO 3

08

Der Koeffizient von Y_1

1

Kennzeichnung der Stelle mit einem Label, wo
der richtige Koeffizient für Y eingefügt wird

2nd Lbl 0
STO 4

11

Berechnung von Y, Multiplikation mit dem
richtigen Koeffizienten und Summierung des
Resultates in Speicher 3

SBR 1 X
RCL 4 = SUM 3

16

Nächster x-Wert in Speicher 2

RCL 1 SUM 2

18

Die Berechnung wird n mal wiederholt.
Ist Speicher Null = 0?

2nd ISZ

19

JA NEIN

Bildung des richtigen Koeffizienten
für die nächste Schleife

GTO 2

20

Berechnung des Wertes Y_{n+1} , Summierung zum
Speicher 3, Multiplikation mit $\frac{w}{3}$ und Anzeige
des Endresultates

SBR 1 SUM 3 RCL 3 X
RCL 5 + 3 = R/S

29

Der zuvor verwendete Koeffizient wird in das
Testregister eingebracht und mit „4“ verglichen

2nd LBL 2
RCL 4 x=1 4

33

Ist 4 in der Anzeige = t?

2nd x=1

34

JA ~ NEIN

Wenn der Koeffizient zuvor 4 war,
ändern Sie ihn auf 2

2

Wenn der Koeffizient zuvor nicht 4 war,
verwenden Sie 4

GTO 0

36

Unterprogramm 1 für die Berechnung von
 $y = f(x)$. (Kann für einen neuen Ausdruck
geändert werden.)

2nd LBL 1
RCL 2 x ²
x ² X RCL 2 - 1 =
3 INV SBR

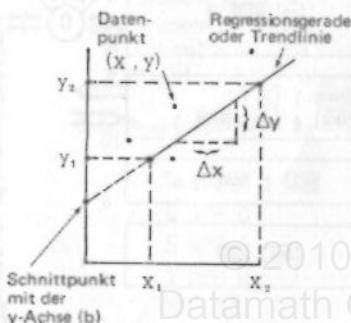
49

Aufheben des Learn-Modus und Rückstellung

LRN RST

LINEARE REGRESSION

Im Schul- oder Berufsleben ist es oft wünschenswert, Vorhersagen über ein künftiges Ereignis zu treffen – die auf Kenntnissen in der vergangenen Entwicklung beruhen. Vielleicht wollen Sie die Auswirkungen auf das Wachstum der Versuchstiere in einem Labor vorherbestimmen, wenn die Ernährung umgestellt wird, oder Sie wollen einen Zusammenhang zwischen zunehmender Werbeaktivität und den Verkaufsziffern eines Produkts feststellen. In jedem Fall wollen Sie eine bekannte Datengruppe als Grundlage nehmen, eine Gerade zeichnen, und dann aus dem Verlauf der Geraden die weitere Entwicklung vorhersagen.



Die übliche Lösung ist, eine Gerade zu zeichnen, die Ihre bekannten Daten oder Testpunkte graphisch optimal wiedergibt. Dann kann man die Gerade in jeder Richtung verlängern und damit gleichartige Ergebnisse voraussagen. Datenpunkte sind im allgemeinen durch 2 Werte bestimmt, der x-Koordinate und der y-Koordinate für jeden aufgezeichneten Punkt.

Die beste Anpassungsgerade für eine Vielzahl von Datenpunkten festzulegen und aufzuzeichnen, ist mühsam und in den meisten Fällen ungenau. Mit Ihrem Rechner können Sie jedoch dieses Programm verwenden, und verschiedene Aspekte des Problems auf einfache Weise lösen.

Zunächst kann man jede Gerade als Gleichung ausdrücken: $y = mx + b$. m ist hier die Steigung der Geraden und b ist ihr Schnittpunkt mit der y-Achse. Die Steigung (m) der Geraden ist das Verhältnis zwischen Δy und Δx und b ist der Punkt, wo die Gerade die y-Achse schneidet. Sobald man Steigung und Schnittpunkt kennt, kann man die Gerade zeichnen. Dann kann man mit der graphischen Darstellung die Gerade verlängern, und sie für Berechnungen und Vorhersagen aufgrund der gegebenen Daten verwenden.

Mit Ihrem Rechner können Sie die Steigung und den Schnittpunkt der Geraden ermitteln, die durch Ihre Datenpunkte geht. Die Berechnung dieser Werte kann jedoch sehr schwierig sein, und in diesem Bereich wirkt sich die Programmierbarkeit sehr positiv aus. Die Formel für die Berechnung der Steigung lautet:

$$m = \frac{\sum_{i=1}^n x_i y_i}{n} - \bar{x}\bar{y}$$
$$\sigma x^2$$

Die Formel für den Schnittpunkt mit der y-Achse:

$$y = \bar{y} - mx$$

n

$\sum_{i=1}^n x_i y_i$ ist die Summe der Koordinaten aller Daten, die von 1 bis n ($x_1 y_1, x_2 y_2, \dots$)

miteinander multipliziert werden. Mit Hilfe einiger spezieller Tasten auf dem Rechner und dem hier entwickelten Programm können Sie m und b mit weit weniger Aufwand berechnen, der sonst für die Verarbeitung derart komplexer Formeln erforderlich ist.

Mit diesem Programm können Sie Ihre Daten auch für Vorhersagen verwenden. Nach Berechnung von m und b gibt man einen neuen „x“-Wert ein, und erhält mit der Formel $y = mx + b$ den entsprechenden „vorhergesagten“ „y“-Wert auf der Geraden. Oder, Sie geben umgekehrt einen „y“-Wert ein, und Sie erhalten den entsprechenden „x“-Wert mit der Formel:

$$x = \frac{y - b}{m}$$

Das Programm ermittelt auch den Bestimmtheitskoeffizienten r^2 . Die Zahl r (die aus r^2 mit der Taste \sqrt{x} berechnet wird), ist der absolute Wert des Korrelationskoeffizienten. Dies ist ein Maß, wie gut die Datenpunkte der Geraden entsprechen oder korreliert sind. Damit kann man die Genauigkeit der getroffenen Vorhersagen bestimmen. Ein Wert nahe 1 weist auf eine hohe Korrelation hin. (Der Rechner könnte Ihre Datenpunkte genau an eine Trendlinie anpassen.) Ein Wert nahe 0 bedeutet, daß die Daten eine niedrige Korrelation haben, und daß die vom Rechner entwickelte Trendlinie keine hohe Zuverlässigkeit für Vorhersagen bietet.

Die Formel für r^2 :

$$r^2 = \frac{m^2 \sigma_x^2}{\sigma_y^2}$$

Das Programm, das hier erstellt wird, enthält viele Berechnungen und schöpft die Kapazität Ihres Rechners aus. Der wesentliche Aspekt ist, daß Ihr Rechner auch schwierige mathematische Probleme verarbeiten kann, daß Sie ein so wertvolles Hilfsmittel wie die lineare Regression für Planung und Voraussagen heranziehen können, ohne daß Sie durch die Vielzahl der Rechengänge unterbrochen werden.



Ziel

Das Programm soll so geschrieben werden, daß man alle Datenpunkte leicht eingeben und dann m , b und r^2 berechnen kann. Schließlich soll es möglich sein, einen y -Wert für einen gegebenen x -Wert zu ermitteln, oder für einen gegebenen y -Wert den entsprechenden x -Wert zu finden. Sie gehen wie folgt vor:

- Eingabe der Datenpunkte: man gibt x ein, drückt **[Σ :t]**, gibt y ein und drückt **[R/S]**. Dieser Vorgang wiederholt sich, bis die Eingabe aller (x, y) Datenpunkte abgeschlossen ist.
- Zur Berechnung von m drückt man **[SBR] 0**
- Zur Berechnung von b drückt man **[SBR] 1**
- Zur Berechnung von r^2 drückt man **[SBR] 2** und dann **[\sqrt{x}]**, um r zu ermitteln. (Denken Sie daran, daß ein Wert für r nahe 1 eine gute Korrelation bedeutet, während ein Wert nahe 0 ein Hinweis ist, daß der Rechner Ihre Daten nicht zuverlässig mit der Geraden korrelieren konnte.)
- Zur Vorhersage eines y -Wertes auf der Geraden für einen gegebenen x -Wert geben Sie x ein und drücken **[SBR] 3**.
- Zur Vorhersage eines x -Wertes auf der Geraden für ein gegebenes y geben Sie y ein und drücken **[SBR] 4**.

© 2010 Joerg Woerner

Planung des Programms



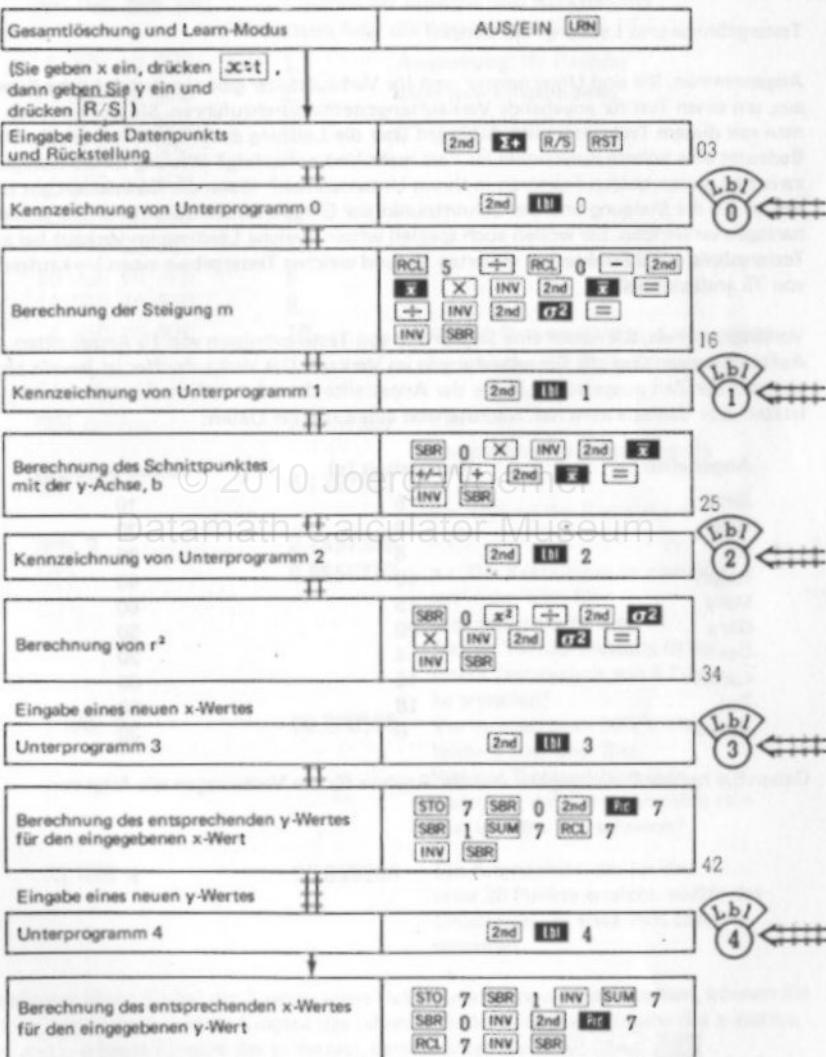
Der erste Schritt ist die Eingabe aller Datenpunkte über die speziellen Tasten für die Zusammenstellung der Daten auf Ihrem Rechner. Diese Tasten sind **[2nd] [Σ :t]**, **[2nd] [σ^2]** und **[2nd] [σ^2]** unten rechts auf der Tastatur. Sie sind am Ende von Kapitel 2 beschrieben.

Anstatt **[2nd] [Σ :t]** für jede Eingabe zu drücken, wird in das Programm eine einfache Folge aufgenommen, mit der sich die Eingabe der Datenpunkte vereinfacht. Für die Berechnung von m , b , r^2 und neuen x - oder y -Werten wird der Rest des Programms in 5 Unterprogramme gegliedert. Verfolgen Sie den Programmablaufplan sehr sorgfältig, weil durch Anwendung von Speicherarithmetik Programmschritte eingespart wurden.

Geben Sie das Programm sorgfältig nach dem Programmablaufplan auf der nächsten Seite ein.

LINEARE REGRESSION

Programmablaufplan



Anmerkung: Sie brauchen den Learn-Modus nicht aufzuheben, da alle 50 Programmschritte verwendet werden und der Rechner den Learn-Modus in diesem Fall automatisch abschaltet.

Ablauf des Programms:

Testergebnisse und Leistung: ein Beispiel

Angenommen, Sie sind Unternehmer und Ihr Verkaufsleiter gibt eine beträchtliche Summe aus, um einen Test für angehende Verkaufsstellte durchzuführen. Sie wollen sehen, ob man mit diesem Test tatsächlich Aussagen über die Leistung des Angestellten machen kann. Bedeutet eine höhere Punktzahl im Test mehr Verkaufserfolg? Wie eng ist die Korrelation zwischen diesen beiden Faktoren in Ihrem Unternehmen? Wenn die Korrelation gut ist, wollen Sie die Steigung und den Schnittpunkt der Geraden wissen, und diese Werte für Voraussagen verwenden. Sie wollen auch speziell wissen, welche Leistung im Verkauf bei einem Testergebnis mit 9 Punkten zu erwarten ist, und welches Testergebnis einen Verkaufserfolg von 75 andeuten würde.

Vorausgesetzt sei, Sie haben eine Stichprobe von Testergebnissen von 10 Angestellten, sowie Aufzeichnungen über die Einzelleistungen im Verkauf. Die Verkaufsziffer ist jeweils als Prozentsatz der Zeit ausgedrückt, in der der Angestellte das wöchentliche Umsatzziel im letzten Jahr überschritten hat. Nachstehend eine Liste der Daten:

Angestellter	Testergebnis (x)	Verkaufserfolg (y)
Jerry	5	10
Ross	13	30
Joe	8	30
Ralph	10	40
Mary	15	60
Gary	20	50
Dean	4	20
Carole	16	60
Ted	18	50
Alecia	6	20

Gehen Sie bei der Dateneingabe und der Analyse für die Voraussagen wie folgt vor:



Tasten	Anzeige/Bemerkungen
[INV] [2nd] [Σ] [RST]	Löschen der Speicher und Rückstellung Dann folgt die Dateneingabe
5 [$x\text{-}t$] 10 [R/S]	1. Anmerkung: Ihr Rechner
13 [$x\text{-}t$] 30 [R/S]	2. zählt jede Eingabe eines
8 [$x\text{-}t$] 30 [R/S]	3. Datenpunkts mit.
10 [$x\text{-}t$] 40 [R/S]	4.
15 [$x\text{-}t$] 60 [R/S]	5.
20 [$x\text{-}t$] 50 [R/S]	6.
4 [$x\text{-}t$] 20 [R/S]	7.
16 [$x\text{-}t$] 60 [R/S]	8.
18 [$x\text{-}t$] 50 [R/S]	9.
6 [$x\text{-}t$] 20 [R/S]	10. Alle 10 Datenpunkte sind eingegeben Berechnung der Steigung
[SBR] 0	2.6837607 = m Berechnung des Schnittpunkts
[SBR] 1	6.1367521 Berechnung der Korrelation
[SBR] 2	0.7497339 = r^2
[\sqrt{x}]	0.8658717 = r Die Korrelation ist ziemlich gut — der Wert für r liegt genügend nahe an +1. Welche Verkaufsleistung ist bei einem Testergebnis von 9 Punkten zu erwarten?
9 [SBR] 3	30.290598 Vorhersage: etwa 30% Verkaufsleistung über dem Ziel. Welches Testergebnis würde andeuten, daß der Angestellte sein Ziel um 75% überschreitet?
75 [SBR] 4	25.659236 Ein Angestellter, der im Test etwa 26 Punkte erreicht, müßte das Umsatzziel um etwa 75% überschreiten.

Anmerkung: Wenn Sie bei der Eingabe eines Datenpunkts einen Fehler machen, können Sie wie folgt korrigieren: erneute Eingabe des falschen Punktes (erneute Eingabe des x-Wertes, Taste [$x\text{-}t$], erneute Eingabe des y-Wertes), dann drücken Sie **[INV] [2nd] [$\Sigma+$]**. Anschließend wird die Eingabe mit korrekten Datenpunkten fortgesetzt.

Wie oft müssen Sie raten, um eine geheime Zahl zwischen 0 und 1023 herauszufinden? Nach jedem Versuch erhalten Sie eine Auswertung (Sie erfahren, ob Ihr Versuch zu hoch oder zu niedrig war). Dieses Programm ist ein Spiel mit Ihrem Rechner, wo Sie beim Autofahren oder beim Camping und überall eine lustige Unterhaltung haben.

So wird gespielt

Das „Hi-Lo“-Programm verwendet eine Anfangseingabe (kleiner als 1), um die Geheimzahl zu erzeugen. Bei jedem Versuch wird +1 angezeigt, wenn Sie zu hoch geraten haben, und -1, wenn die geratene Zahl zu niedrig ist, oder die Geheimzahl blinkt in der Anzeige, wenn Sie richtig getippt haben. Nach Eingabe des Programms gehen Sie wie folgt vor:

- Drücken Sie die Taste **[•]**, und geben Sie dann die Uhrzeit, die Nummer des Monats und den Tag des Monats ein. (Damit haben Sie eine geeignete Anfangszahl kleiner als 1.) Jetzt drücken Sie **[R/S]**. An dieser Stelle weiß der Rechner die Geheimzahl und zeigt eine Null an – das Spiel kann beginnen.
- Versuchen Sie nun, die Zahl zu erraten, geben Sie Ihren Versuch ein und drücken Sie **[SBR] 4**. Ist Ihr Versuch zu hoch, wird 1 angezeigt. Ist er zu niedrig, erscheint -1 in der Anzeige. Wenn Sie richtig geraten haben, blinkt die Geheimzahl.
- Machen Sie noch einen Versuch, geben Sie die neue Zahl ein und drücken Sie **[SBR] 4**.
- Nachdem Sie die Zahl erraten haben, drücken Sie **[RCL] 1**, um die Anzahl Ihrer Versuche abzulesen. Für ein neues Spiel drücken Sie **[CLR]**, um das Blinken abzustellen, und **[SBR] 1** als Anweisung für den Rechner, eine neue Geheimzahl zu erzeugen. Dann geben Sie wieder Ihre Versuche ein und drücken jeweils **[SBR] 4**.

Tasten Sie Ihr Programm nach der Auflistung ein. (Vergessen Sie nicht, den Rechner aus- und wieder einzuschalten und drücken Sie **[LRN]**.)



Wichtig: Das Programm kann nur eingeschaltet werden, wenn es sich auf einer speziell geschaffenen Anwendung befindet. Wenn es auf einer anderen Anwendung läuft, kann es nicht eingeschaltet werden. Um dies zu verhindern, müssen Sie sicherstellen, dass es sich auf einer solchen Anwendung befindet. Wenn es auf einer anderen Anwendung läuft, kann es nicht eingeschaltet werden. Um dies zu verhindern, müssen Sie sicherstellen, dass es sich auf einer solchen Anwendung befindet.

HI-LO ZAHLENRATESPIEL

Hi-Lo – Zahlenratespiel

Taste	Speicher- platz	Kode	Taste	Speicher- platz	Kode
AUS-EIN [LRN]			[STO] 7	23	32 7
[STO] 0	00	32 0	[CLR]	24	15
[2nd] [BL] 1	01	86 1	[R/S]	25	81
0	02	00	[2nd] [BL] 4	26	86 4
[STO] 1	03	32 1	[STO] 2	27	32 2
[2nd] [BL]	04	30	1	28	01
+	05	75	[SUM] 1	29	34 1
[RCL] 0	06	33 0	[RCL] 2	30	33 2
=	07	85	[INV] [2nd] [x=1]	31	-76
[Y ^x]	08	35	[GTO] 8	32	51 8
8	09	08	[2nd] [x=1]	33	66
-	10	65	[GTO] 7	34	51 7
[2nd] [BL]	11	49	1	35	01
=	12	85	[=]	36	85
[STO] 0	13	32 0	[R/S]	37	81
[X]	14	55	[2nd] [BL] 8	38	86 8
1	15	01	1	39	01
0	16	00	[+/-]	40	84
2	17	02	[=]	41	85
3	18	03	[R/S]	42	81
+	19	75	[2nd] [BL] 7	43	86 7
1	20	01	[RCL] 7	44	33 7
[=]	21	85	[GTO] 3	45	51 3
[2nd] [BL]	22	49	[LRN] [RST]		

Ablauf des Programms

Hier ein typischer Spielverlauf:



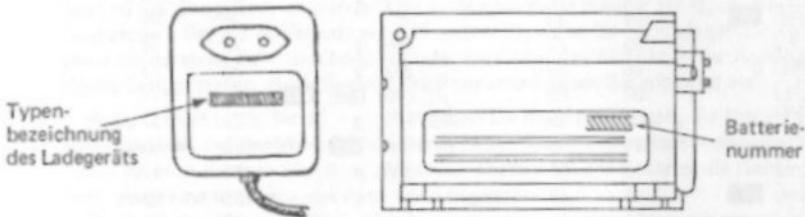
Taste	Anzeige/Bemerkungen
	Eingabe einer Anfangszahl (kleiner als 1) nach Ihrer Wahl
.243411 [RST]	0 Die Geheimzahl ist bereit
500 [SBR] 4	1. Versuch zu hoch
250 [SBR] 4	-1. Versuch zu niedrig
400 [SBR] 4	-1. Versuch zu niedrig
450 [SBR] 4	-1. zu niedrig
475 [SBR] 4	1. zu hoch
470 [SBR] 4	1. zu hoch
466 [SBR] 4	466. Blinken – Sie haben richtig geraten
[CLR] [RCL] 1	7. Sie benötigten 7 Versuche

Für ein erneutes Spiel drücken Sie [CLR] [SBR] 1. Dann geben Sie Ihren Versuch ein, drücken [SBR] 4 und setzen das Spiel fort.

ANHANG A: BATTERIE- UND NETZBETRIEB

Normalbetrieb

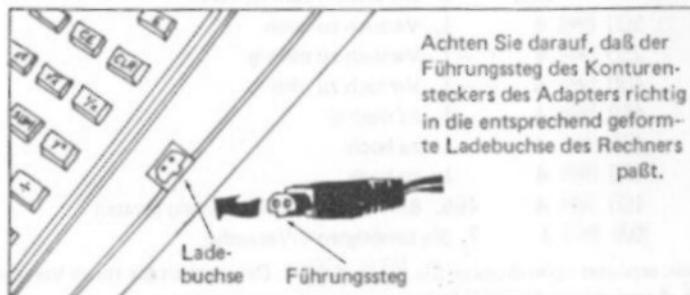
Ihr Rechner kann netzunabhängig betrieben werden, wenn das Batteriepaket von Zeit zu Zeit mit dem mitgelieferten Netzadapter/Ladegerät aufgeladen wird. Es ist wichtig, das richtige Adapter/Ladegerät zu verwenden. Wenn Batteriepaket oder Ladegerät ausgetauscht werden müssen, achten Sie darauf, daß Sie den richtigen Ersatz bekommen.



Anmerkung: Sie können mit Ihrem Rechner zwei verschiedene Kombinationen aus Batteriepaket und Netzadapter/Ladegerät verwenden: das Batteriepaket BP 6 zusammen mit dem AC 9900 H Netzadapter/Ladegerät, oder das Batteriepaket BP 7 mit dem AC 9900 R Adapter/Ladegerät.

Vorsicht: Die Anwendung eines ungeeigneten Adapter/Ladegeräts kann Ihrem Rechner eine falsche Spannung zuführen und das Gerät beschädigen.

Um mit dem Rechner eine maximale Zeitspanne netzunabhängig arbeiten zu können, schließen Sie das Adapter/Ladegerät an eine 220V/50Hz Steckdose an, verbinden damit den Rechner, und laden das Batteriepaket mindestens 4 Stunden bei ausgeschaltetem Rechner oder 10 Stunden bei eingeschaltetem Rechner. Das Adapter/Ladegerät und das Batteriepaket können im Betrieb warm werden. Dies ist jedoch normal und hat keine weiteren Auswirkungen. (Vorsicht: Das Batteriepaket BP 6 darf nur geladen werden, wenn es richtig in den Rechner eingesetzt ist.)



Bei voll aufgeladenem Batteriepaket arbeitet der Rechner etwa 3 bis 4 Stunden, bis ein erneutes Aufladen erforderlich ist. Zögern Sie jedoch nicht, das Adapter/Ladegerät anzuschließen, wenn Sie wissen oder vermuten, daß das Batteriepaket fast entladen ist. Ein nahezu entladenes Batteriepaket kann alle Rechneroperationen negativ beeinflussen und Ursache für falsche Ergebnisse sein. Typische Zeichen für ein entladenes Batteriepaket sind eine schwache unkontrolliert aufleuchtende oder völlig dunkle Anzeige.

Die Lebensdauer der Nickel-Cadmium-Zellen des Batteriepakets ist zwar schwer vorauszusagen, aber bei normalem Gebrauch halten die Batterien etwa zwei bis 3 Jahre oder etwa 500 bis 1000 Ladezyklen.

Periodisches Aufladen

Obwohl der Rechner unbegrenzt mit angeschlossenem Adapter arbeitet, kann das aufladbare Batteriepaket mit der Zeit seine Speicherkapazität verlieren, wenn es nicht gelegentlich entladen wird. Im Interesse einer langen Lebensdauer der Batterien empfiehlt es sich, den Rechner mindestens zweimal im Monat netzunabhängig zu betreiben, die Batterien dabei fast völlig zu entladen und dann entsprechend wieder aufzuladen.

Tief-Entladen der Batterien

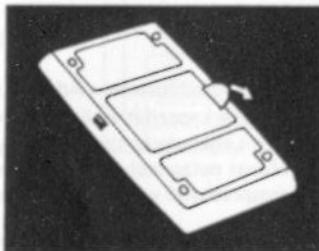
Bleibt der Rechner längere Zeit eingeschaltet, wenn das Batteriepaket bereits entladen ist (etwa zufällig über Nacht), schließen Sie das Netzadapter/Ladegerät mindestens 24 Stunden lang bei ausgeschaltetem Rechner an. Wenn danach der normale Batteriebetrieb nicht mehr aufgenommen werden kann, muß das Batteriepaket ausgetauscht werden. Wiederholtes restloses Entladen zerstört auf die Dauer das Batteriepaket. Ersatz- und Austausch-Batteriepakete erhalten Sie bei Ihrem Händler oder direkt von Texas Instruments.

Lagerung

Wenn der Rechner mehrere Wochen lang gelagert ist oder nicht benutzt wird, muß das Batteriepaket wahrscheinlich vor netzunabhängigem Betrieb geladen werden. Das Batteriepaket ist auslaufgeschützt; deshalb ist es ungefährlich, wenn der Rechner mit eingesetztem Batteriepaket aufbewahrt wird.

Austausch des Batteriepakets

Das Batteriepaket läßt sich schnell und einfach ausbauen. Halten Sie den Rechner mit den Tasten nach unten. Stecken Sie eine kleine Münze in den Schlitz an der Unterseite des Rechners. Nach einer leichten Kippbewegung springt die Abdeckung auf. Trennen Sie sorgfältig die Verbindung zwischen Batteriepaket und Rechner. Dann kann das Batteriepaket ganz herausgenommen werden.



© 2010 Joerg Woerner

Die Metallkontakte am Batteriepaket für Ladegerät und Rechner sind die Batterieanschlüsse. Achten Sie immer darauf, daß kein Metallgegenstand mit diesen Anschlüssen in Berührung kommt und die Batterien kurzschließt.

Beim Einsetzen des Batteriepakets stecken Sie zunächst die Verbindungsdrähte an die Anschlüsse des Batteriepakets. Das Einsticken ist kein Problem, weil die Verbindung nur in einer Position paßt. Dann setzen Sie das Batteriepaket so in die Öffnung, daß der kleine Vorsprung an dem einen Ende unter die Kante des Rechnerbodens paßt. Mit geringem Druck rastet das Batteriepaket in die richtige Position. (*Wenden Sie keine Gewalt an. Das Batteriepaket schnappt leicht ein, wenn es richtig geführt ist.*)

ANHANG B: ABHILFE BEI STÖRUNGEN

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. Sind die vorgeschlagenen Abhilfen unwirksam, wenden Sie sich schriftlich oder in dringenden Fällen auch telefonisch an Texas Instruments (siehe „Wenn Sie Fragen haben oder Unterstützung brauchen“ in diesem Anhang). Bitte geben Sie eine genaue Beschreibung der Fehlersymptome.

Störung	Abhilfe
1. Aus einem nicht ersichtlichen Grund bleibt die Anzeige dunkel.	Drücken Sie R/S und halten Sie die Taste kurz fest. Wenn jetzt wieder eine Anzeige erfolgt, verarbeitete der Rechner gerade ein langes Programm oder befand sich in einer fortlaufenden Programmschleife.
2. In der Anzeige erscheinen unrichtige Ergebnisse, unsinnige Zahlen blinken, sie wird dunkel oder verschwindet ganz.	Das Batteriepaket kann entladen oder nicht richtig eingesetzt sein. Prüfen Sie auch, ob der EIN-AUS-Schalter ganz auf ON gestellt ist. Wahrscheinlich ist das Batteriepaket entladen oder nicht richtig eingesetzt. Siehe „Batterie- und Netzbetrieb“ in Anhang A.
3. Die Anzeige blinkt, während Operationen über die Tastatur durchgeführt werden.	Eine unzulässige Operation oder Tastenfolge wurde gefordert oder die Grenzen des Rechners wurden überschritten. Eine Liste dieser Bedingungen siehe Anhang C.

Wenn keines der obigen Verfahren die Störung beseitigen kann, senden Sie Rechner und Ladegerät an das zuständige Texas Instruments Service-Center.

Versandart:

BRD: Registriertes und versichertes Paket
A: Einschreibe-Sendung
CH: Eingeschriebenes Postpaket

Für Reparaturen außerhalb des Gewährleistungszeitraums werden Sie mit den zur Zeit der Einsendung geltenden Servicekosten belastet. Schildern Sie bitte schriftlich Ihre Schwierigkeiten mit dem Rechner und legen Sie Ihre vollständige Anschrift mit Name, Straße und Wohnort mit Postleitzahl bei. Schicken Sie die Sendung sorgfältig verpackt zum Schutz gegen rauhe Behandlung an eine der Texas Instruments Niederlassungen, die auf dem Rückumschlag der Bedienungsanleitung aufgeführt sind.

ANHANG B: ABHILFE BEI STÖRUNGEN

Wenn Sie Fragen haben oder Unterstützung brauchen

Wenn Sie Fragen zur Rechnerreparatur, zum Zubehörkauf oder zu den Funktionen des Rechners haben, wenden Sie sich bitte schriftlich und nur in dringenden Fällen telefonisch an die nächstgelegene Texas Instruments Kundendienstabteilung. Hier werden auch Ihre technischen Fragen zur Programmierung, zu speziellen Anwendungsgebieten usw. beantwortet.

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".

Datamath Calculator Museum

ANHANG C: FEHLERBEDINGUNGEN

Eine blinkende Anzeige ist der Hinweis, daß die Grenzen des Rechners überschritten wurden, oder daß eine unzulässige Operation gefordert wurde. Mit den Tasten **[CE]** oder **[CLR]** kann man das Blinken abstellen. **[CLR]** löscht auch die Anzeige und unvollständige Operationen. **[CE]** stoppt nur das Blinken, und weitere Berechnungen mit nicht beeinflußten unvollständigen Operationen sind möglich. Die Anzeige blinkt aus folgenden Gründen:

1. Eingabe oder Ergebnis (in die Anzeige oder in den Speichern) liegt außerhalb des Rechenbereichs $\pm 1 \times 10^{-99}$ bis $\pm 9.9999999 \times 10^{99}$. Der überschrittene Grenzwert blinkt und zeigt Kapazitätsüberlauf bzw. -unterschreitung an.
2. Umkehrung einer trigonometrischen Funktion mit einem unzulässigen Wert für die Funktion, wie bei $\sin^{-1}x$ für x größer 1. Der unzulässige Wert x blinkt.
3. Wurzel oder Logarithmus einer negativen Zahl. Als Hinweis auf den Vorzeichenfehler blinkt die Wurzel oder der Logarithmus des Absolutwerts der Zahl.
4. Potenz oder Wurzel einer negativen Zahl. Die Potenz (oder die Wurzel) des Absolutwerts der Zahl blinkt.
5. Zwei Operationstasten werden nacheinander gedrückt. Davon betroffen sind die Tasten **[+]**, **[-]**, **[X]**, **[÷]**, **[y^x]** und **[INV]** **[y^x]** (für $\sqrt[3]{y}$). Die zuletzt eingegebene Zahl blinkt.
6. Die zuletzt eingegebene Zahl blinkt auch dann, wenn man **=** oder **)** nach **+**, **-**, **[X]**, **[÷]** oder **[INV]** **[y^x]** (für $\sqrt[3]{y}$) drückt.
7. Wenn mehr als 9 linke Klammern geöffnet oder mehr als 4 Operationen unvollständig sind. Die 10te Klammer oder die 5te Operation werden nicht akzeptiert, die Berechnung kann also fortgesetzt werden. Die zuletzt angezeigte Zahl blinkt.
8. Wenn man eine Zahl durch Null dividiert, blinkt in der Anzeige der Wert "9.9999999 99" (Ausnahme ist 0 **÷** 0 **=** – in diesem Fall blinkt 1.)
9. Eine Operationstaste wird vor Abschluß einer Speicheroperation, einer Festkommaeinstellung oder einer direkten Sprungoperation gedrückt. Der Wert in der Anzeige blinkt (z.B. **STO** **+**).
10. Der Versuch, Speicherung, Aufruf oder andere Speicheroperationen mit anderen Adressen als 0 bis 7 zu ergänzen (z.B. **STO** 8).
11. Der Versuch, Funktionen mit Eingabewerten zu ermitteln, die nicht in den folgenden Definitionsbereichen liegen:

Funktion	Definitionsbereich
$\sin^{-1}x$, $\cos^{-1}x$	$-1 \leq x \leq 1$
e^x	$-227.95592 \leq x \leq 230.25850$
10^x	$-99 < x < 100$
$\ln x$	$1 \times 10^{-99} \leq x < 1 \times 10^{100}$
$\log x$	$1 \times 10^{-99} \leq x < 1 \times 10^{100}$
[INV] [y^x]	$1 \times 10^{-50} < R$

-
12. Direkte Sprungbefehle (**GTO** oder **SBR**), die zu nicht zugeordneten Labelpositionen verzweigen sollen. Der augenblickliche Anzeigewert blinkt. Eine Fehlerbedingung ergibt sich auch dann, wenn bei der Tastenfolge **GTO** **[2nd]** nn die Adresse nn größer als 49 ist.
 13. Der Versuch, in einem Programm über den Speicherplatz 49 hinaus zu gehen. Der augenblickliche Anzeigewert blinkt.
 14. Der Versuch, X oder σ^2 zu berechnen, wenn nur ein einzelner Datenpunkt oder wenn keine Datenpunkte eingegeben wurden.

Fehler beim Programmablauf

Tritt einer der aufgeführten Fehler in einem Programm auf, wird es unterbrochen und die Anzeige blinkt . D.h., Sie haben in einem Programm eine Operation gefordert, die der Rechner selbst im Rechenmodus nicht durchführen kann.

© 2010 Joerg Woerner

Datamath Calculator Museum

ANHANG D: ANGEZEIGTE ERGEBNISSE UND GENAUIGKEIT

Grundsätzlich wird die mathematische Toleranz des Rechners durch die Anzahl der Stellen gesteuert, die für die Berechnungen verwendet werden. Der Rechner arbeitet scheinbar mit 8 Stellen, wie sie in der Anzeige ausgewiesen sind, tatsächlich werden aber alle Berechnungen mit 11 Stellen ausgeführt. In Verbindung mit der eingebauten 5/4-Rundung sichern diese zusätzlichen Stellen die 8-stellige Anzeige und erhöhen so die Genauigkeit. Betrachten Sie das folgende Beispiel, wenn die Schutzstellen fehlen:

$$1/3 \times 3 = .99999999 \text{ (ungenau)}$$

Dieses Beispiel zeigt, daß $1 : 3$, mit 3 multipliziert, ein ungenaues Ergebnis erzeugt. Bei einer 11-stelligen Neuner-Reihe dagegen wird der Wert bei der Rundung auf 8 Stellen auf 1 gerundet.

Die mathematischen Funktionen höherer Ordnung verwenden iterative Berechnungen. Im allgemeinen setzt sich der kumulative Rundungsfehler nach der 8-stelligen Anzeige fort, so daß keine Auswirkungen erkennbar sind. Die drei Schutzstellen sichern den Anzeigewert gegen kleine kumulative Fehler.

Normalerweise kann man diese Schutzstellen völlig außer acht lassen. Bei bestimmten Berechnungen können sie jedoch als unerwartetes Ergebnis in Erscheinung treten. Die Ergebnisse zweier Berechnungen können in der Anzeige gleich sein, aber in ihren Schutzstellen Unterschiede aufweisen. Wenn diese beiden Ergebnisse subtrahiert werden, kann der Rechner dann ein Ergebnis ungleich Null anzeigen.

Diese Tatsache hat besondere Bedeutung, wenn Sie eigene Programme schreiben. Wenn Sie ein Rechenergebnis testen, ob es mit einem anderen Wert gleich ist, zum Beispiel mit dem **2nd [x=]** Befehl, müssen Sie Vorsichtsmaßnahmen treffen, um falsche Berechnungen wegen der Unterschiede in den Schutzstellen zu vermeiden. Die Folge **[EE] [INV] [EE]** entfernt die Schutzstellen eines Resultats und nur der gerundete Anzeigewert wird weiterverwendet.

Im Standardformat der Anzeige sind die Ergebnisse für alle Berechnungen genau, sofern nicht die in Anhang C aufgelisteten Einschränkungen übertreten wurden. Hierzu gelten folgende Ausnahmen:

Trigonometrische Funktionen

Alle angezeigten Stellen sind für den Bereich ± 36000 Altgrad ($\pm 200 \pi$ Radian und ± 40000 Gon) genau. Im allgemeinen nimmt die Genauigkeit außerhalb des angegebenen Bereichs um eine Stelle je Dekade ab. Eine Ausnahme bildet der Tangens eines ungeraden Vielfachen von $\pm 90^\circ$, $\pm \pi/2$ Radian oder ± 100 Gon, der zu einer Überlaufbedingung führt, weil die Funktion an diesen Punkten nicht definiert ist.

Potenzen und Wurzeln

Ein Genauigkeitsverlust für Potenzen und Wurzeln kann dann entstehen, wenn die Basis y sehr nahe an 1 geht und die Potenz x sehr groß wird.

ANHANG E: TASTENKODES

Rechner-Tastenkodes – nach der Lage auf der Tastatur

Reihen

1	kein 2nd Code	Minus Zeichen	I 18	C 19	CLR 15
2	LN 25 LRN kein Code	P+I 27 x±I 22	SIN 28 X² 23	COS 29 LN 24	TAN 20 1/X 25
3	Fwd 36 SST kein Code	BS kein Code STO 32	EE 38 RCL 33	Pd 39 SUM 34	BT 30 Y* 35
4	No 46 BST kein Code	BS kein Code EE 42	EE 48 I 43	BT 49 Y 44	IxI 40 ÷ 45
5	GTO 51	Z 07	8 08	9 09	BT 50 X 55
6	SBR 61	4 04	5 05	6 06	Rd 60 - 65
7	RST 76	1 01	2 02	3 03	Rd 70 + 75
8	BL 86 R/S 81	0 00	Z+ 88 * 83	X 89 +/- 84	BT 80 = 85
Spalten 1 (für → 6 Zweitfunktionen)		2 7	3 8	4 9	5 0

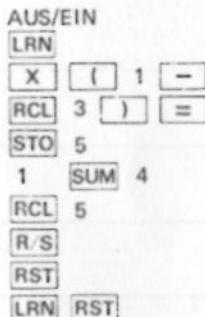
Rechner-Tastenkodes in numerischer Reihenfolge

00	0	- 27	INV	2nd	P+I	50	2nd	Sig
01	1	- 28	2nd	SIN		51	GTO	
02	2	- 28	INV	2nd	SIN	55	X	
03	3	- 29	2nd	EE		56	2nd	Dz
04	4	- 29	INV	2nd	EE	56	INV	2nd
05	5	- 30	2nd	BT		60	2nd	Rd
06	6	- 32	STO			61	SBR	
07	7	- 33	RCL			61	INV	SBR
08	8	- 34	SUM			65	-	
09	9	- 34	INV	SUM		66	2nd	x±I
13	IxI	- 35	Y*			66	INV	2nd
- 13	INV	- 35	INV	Y*		70	2nd	BTd
14	CE	- 36	2nd	Pause		71	RST	
15	CLR	- 36	2nd	Ex		75	+	
18	2nd	- 39	2nd	Pd		76	2nd	x±I
- 18	INV	- 39	INV	2nd				
- 18	2nd	- 39	2nd	Pd		- 76	INV	2nd
19	2nd	- 40	2nd	IxI		- 80	2nd	BT
- 19	INV	- 40	2nd	IxI		- 80	2nd	BT
20	2nd	- 42	EE			81	R/S	
- 20	INV	- 42	INV	EE		83	*	
22	x±I	- 43	I			84	+/-	
23	x²	- 44	Y			85	=	
24	LN	- 45	EE			86	2nd	BL
25	LN	- 45	2nd	Nop		86	2nd	Z+
26	2nd	- 48	2nd	Fix		88	2nd	BT
- 26	INV	- 48	INV	2nd				
27	2nd	- 49	INV	2nd				
- 27	INV	- 49	INV	2nd				

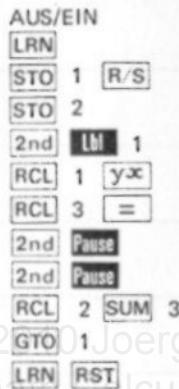
LÖSUNGSVORSCHLÄGE FÜR DIE MODIFIZIERTEN AUFGABEN BEI DEN ANWENDUNGSBEISPIELEN

Anmerkung: Im allgemeinen kann eine Aufgabe immer auf verschiedene Weise programmiert werden. Hier jeweils eine mögliche Lösung.

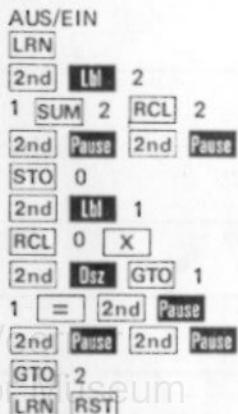
Im Handel
Seite 4 - 1

AUS/EIN
LRN


Abzählen
Seite 4 - 3

AUS/EIN
LRN


Fakultät
Seite 4 - 6

AUS/EIN
LRN


© 2000 Joerg W.

47.0538 2nd B3
INV 2nd B3

47.093889
47.0538

Um wieder zu dieser Wiederaufgabe zurück zu gelangen, muss man die Tastenfolge 2nd B3 abdrücken und dann die Tastenfolge INV 2nd B3 drücken. Das Ergebnis ist dann 47.0538. Um wieder zu der vorherigen Wiederaufgabe zurück zu gelangen, muss man die Tastenfolge 2nd B3 abdrücken und dann die Tastenfolge INV 2nd B3 drücken. Das Ergebnis ist dann 47.093889.

Um wieder zu einer anderen Wiederaufgabe zurück zu gelangen, muss man die Tastenfolge 2nd B3 abdrücken und dann die Tastenfolge INV 2nd B3 drücken. Das Ergebnis ist dann 47.0538. Um wieder zu einer anderen Wiederaufgabe zurück zu gelangen, muss man die Tastenfolge 2nd B3 abdrücken und dann die Tastenfolge INV 2nd B3 drücken. Das Ergebnis ist dann 47.093889.

Alle Rechte vorbehalten. Das Material darf nicht kopiert, heruntergeladen oder weiterverbreitet werden. Es darf nur für den persönlichen Gebrauch heruntergeladen und verwendet werden. Es darf nicht für kommerzielle Zwecke benutzt werden.

Autoren: Joerg Woerner
Redaktion: Joerg Woerner
Illustrationen: Joerg Woerner
Fotos: Joerg Woerner
Layout: Joerg Woerner

© 2010 Joerg Woerner
Datamath Calculator Museum

Wichtig!

Texas Instruments gibt in Bezug auf dieses Material weder ausdrücklich noch stillschweigend Gewährleistung, einschließlich, aber nicht nur begrenzt, auf stillschweigend vorausgesetzte Gewährleistung durchschnittlicher Qualität und Eignungsgarantie für einen bestimmten Zweck, und stellt dieses Material ausschließlich auf der Basis des gegenwärtigen Zustands zur Verfügung.

Entstehende, für Neben- oder Folgeschäden in Verbindung mit oder als Konsequenz aus dem Kauf oder der Anwendung dieses Materials, und die einzige und ausschließliche Haftung von Texas Instruments, ohne Rücksicht auf die Prozessform, wird nicht über den Kaufpreis dieses Buches hinausgehen.

© 2010 Joerg Woerner
Datamath Calculator Museum

GB

KEEP THE CARD ON THE REVERSE SIDE
(PROOF OF PURCHASE DATE)

- IMPORTANT : The only applicable warranty conditions are those described in your Owner's Manual.

Please fill in the attached card and send it back to your closest Texas Instruments office.
(See address on the back side cover of your owner's manual).

F

CONSERVEZ AVEC SOIN LA CARTE IMPRIMEE AU DOS
(JUSTIFICATION DE LA DATE D'ACHAT)

- IMPORTANT : Les seules conditions de garantie applicables sont celles décrites dans votre manuel d'utilisation.

Veuillez remplir la carte ci-contre et la renvoyer au bureau le plus proche de Texas Instruments (adresse sur le dos de la couverture de votre manuel d'utilisation).

D

BEWAHREN SIE DIE RÜCKSEITIGE KARTE SORGFÄLTIG AUF.
(NACHWEIS ÜBER DAS KAUFDATUM)

- WICHTIG : Es gelten nur die in Ihrer Gebrauchsanweisung abgedruckten Gewährleistungsbedingungen.

Bitte die abhängende Karte sorgfältig ausfüllen und sofort an die für Sie zuständige Texas Instruments Niederlassung schicken (Adresse siehe Rückseite Bedienungsanleitung).

CONSERVATE CON CURA LA SCHEDA STAMPATA SUL RETRO.
(PROVA DELLA DATA DI ACQUISTO).

- IMPORTANTE : Le sole condizioni di garanzia valide sono quelle descritte sul vostro libretto d'istruzioni.

Preghiamo riempire la scheda allegata e spedirla al più vicino ufficio della Texas Instruments indicato sul retro della copertina del manuale d'istruzione).

NL

BEWAAR DIE KAART AAN OMKEZIJDE ZORGVULDIG
(BEWIJS VAN AANKOOPDATUM).

- BELANGRIK : Van toepassing zijn uitsluitend de in uw gebruiksaanwijzing vermelde garantieverwoeders.

Colore da angiographie haert, volledig ingevuld, op te zenden aan het voor u dichthabende kantoor van Texas Instruments (na adresven op achterzijde van uw gebruiksaanwijzing).

© 2010 Joerg Woerner
Datamath Calculator Museum

D

TEXAS INSTRUMENTS

GEWÄHRLEISTUNGSKARTE ~ SCHEDA DI GARANZIA

Modell

TI-57

Modell

Serien Nr.

Datum :

Data :

1 Herr, Sig. re2 Frau, Fräulein, Sig. ra, Sig. na

Numero di serie

3 Firma, Ditta

Familienname

Cognome

Vorname

Nome

Firma :

Ditta :

Strasse :

Indirizzo :

Ort :

Città :

Postleitzahl :

Codice postale :

Land :

Paese :

War der Rechner ein Geschenk ?

Ja

Nein

No

Gebrauch

1 2

Privat

Personale

3

Geschäftlich

Professionale

4

Beides

Ambiduale

La calcolatrice è stata regalata ?

Ja

Nein

No

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

F

TEXAS INSTRUMENTS

WARRANTY CARD • CARTE DE GARANTIE

GEWAARLEISTUNGSKARTE • GARANTIEKAART

PROOF-OF-PURCHASE DATE
JUSTIFICATIF DE LA DATE D'ACHAT
NACHWEIS ÜBER DAS KAUFDATUM
PROVA DELLA DATA DI ACQUISTO
AANKOOPDATUMBEWIJS

Retailer Stamp Cachet du revendeur Handler Stempel Timbro del rivenditore Stempel wederverkoper	Date, Date, Datum, Data, Datum. <i>21.3.79</i>
<i>deca</i> Lebensmittelvertrieb LICHETZ & Co. 2000 HAMBURG 75 Averhohmstraße 24, Tel. 040 722 36 02	

Model TI-57 Serial N°
Modèle Modell Serie Nr.
Modell Modello Numero di serie
Model Model Serienummer

1 MR. Monsieur
2 Miss, Mrs. Mme, Melle,
Monsieur
Herr
Sig. re
Dhr.

3 Company
Société
Firma
Firma
Firma
Firma

Last Name Nom Familienname Vorname
Cognome Nome Achternaam Voornaam

First Name
Prénom
Vorname
Nome
Voornaam

Address
Adresse
Adresse
Indirizzo
Adres

Town Ville Ort
Città Stad

P. O. Code
Code Postal
Postleitzahl
Codice Postale
Postcode

Country
Pays
Land
Paese
Land

Please describe failure
Description de la panne
Bitte, beschreiben Sie den gefundenen Fehler
Descripción del guasto
Korte beschrijving van de storing



KEEP

CONSERVEZ

BEWAHREN SIE

CONSERVATE

BEWAREN

© 2010 Joerg Woerner
Datamath Calculator Museum



WARRANTY
CARD



CARTE
DE GARANTIE



GARANTIE
KAART

TEXAS INSTRUMENTS

Model TI-57	Serial N° N° de série Serie nr.	Date : Date : Datum :
1 <input type="checkbox"/> Mr. Monsieur, Dhr.	2 <input type="checkbox"/> Miss, Mrs. Mme, Melle, Mevr., Mej.	3 <input type="checkbox"/> Company, Société, Firma
Last Name Nom Achternaam	First Name Prénom Vorname Voornaam	
Address : Adresse : Adres :		
Town : Ville : Stad :	P.O. Code : Code Postal : Postleitzahl : Codice Postale : Postcode :	Country : Pays : Land : Paese : Land :

Was calculator a gift ? Yes No
Was dit een cadeau ? Ja Nee
Was rekenmachine een kadootje ? Ja Nee

Personal
Personnelle
Persoonlijk
2 Professional
Berouwspalieve
3 Both
Les deux
Beide

WHERE PURCHASED ?
LIEU D'ACHAT ?
WAAR GEKOCHT ?

- A Dept. Store
Grand magasin
Warenhaus
- B Hypermarket
Grande Surface
Hypermarkt
- C Office equipment dealer
Magasin d'équipement de bureau
Kantoorbenodigdheden
- D Calculator shop
Magasin de calculatrices
Specialzaak v. rekenmachines
- E Mail order
Par correspondance
Postorder

- F Home appliance shop
Magasin d'électro-ménager
Winkel huishoudelijke app.
- G Radio-TV shop
Magasin Radio-Télévision
Radio-/Televistenzaak
- H Camera shop
Magasin photo
Fotoshop
- I Stationery shop
Librairie - Papeterie
Boekhandel
- J Other (Specify)
Autre (Préciser)
Elders (Toelichten)

Do you already own a calculator ? Yes No
Possédez-vous une autre calculatrice ? Ja Nee
Besit u al een rekenmachine ? Ja Nee

YOUR OCCUPATION VOTRE PROFESSION UW BEROEP	G <input type="checkbox"/> Teacher, Professor Professeur Onderwijzer Shop keeper Commerçant Winkelier
A <input type="checkbox"/> Engineer Ingenieur Ingénieur	H <input type="checkbox"/> Physician/Lawyer Médecin/Juriste Dokter/Jurist
B <input type="checkbox"/> Scientist Chercheur Wetenschapper	I <input type="checkbox"/> Technician Technicien Technicus
C <input type="checkbox"/> Business man Homme d'affaires Zakenman	J <input type="checkbox"/> Accountant Comptable Boekhouder
D <input type="checkbox"/> Other (Specify) Autre (Préciser) Elders (Toelichten)	K <input type="checkbox"/> Architect/Surveyor Architecte/Géomètre Architect/Landmeter
E <input type="checkbox"/> College student Lyceen Scholler	L <input type="checkbox"/> Financier Financier Financier
F <input type="checkbox"/> University student Etudiant Student	M <input type="checkbox"/> Other (Specify) Autre (Préciser) Andere (toelichten)

YOUR AGE
VOTRE AGE
UW LEEFTIJD

1 <input type="checkbox"/> — 14
2 <input type="checkbox"/> 15 - 18
3 <input type="checkbox"/> 19 - 28
4 <input type="checkbox"/> 29 - 48
5 <input type="checkbox"/> 49 +



EIN JAHR GEWÄHRLEISTUNG

Texas Instruments gewährleistet dem Endverbraucher (Erstkäufer), daß der elektronische Taschenrechner TI-57 (einschließlich Aufladerät) von Texas Instruments bei sachgemäßer Wartung und sachgemäßem Gebrauch für die Dauer von einem (1) Jahr ab Kaufdatum frei ist von Herstellungs- und Materialfehlern.

Der Gewährleistungsanspruch besteht nur, wenn :

1. Der Rechner nicht 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.

Während der Gewährleistungszeit wird der mangelhafte Rechner nach Wahl von Texas Instruments kostenlos repariert oder durch einen einwandfreien nachgebesserten Austauschrechner ("REFURBISHED") entsprechender Qualität und Güte ersetzt, sofern der Rechner portofrei und versichert mit Kaufpreis-datumnachweis an Texas Instruments geschickt wird. Bei berechtigten Gewährleistungsansprüchen kann Erstattung der Versandkosten verlangt werden (nur Bundesrepublik Deutschland).

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

Weitere Ansprüche, insbesondere Ansprüche auf Ersatz von Schaden, die nicht an dem Rechner selbst entstanden sind, sind ausgeschlossen.

WICHTIG ! Bei Rücksendung zwecks Reparatur bitte Versand- und Bedienungshinweise in diesem Buch sorgfältig beachten !

SERVICE CENTERS

BELGIE - BELGIQUE

Avenue Edouard Lacomblé, 21
1040 - Bruxelles Brussel
Tel. (2) 733 96 23

CANADA

41 Shelley Road
Richmond Hill, Ontario
Tel. (416) 889 73 73

DANMARK

Marielundvej 46 E
2730 Herlev
Tel. (02) 91 74 00

DEUTSCHLAND

Kepserstraße 33
8050 - Freising
Tel. (08161) 7411

ENGLAND

Manton Lane
Bedford, MK41 7PU
Tel. (0234) 67466

ESPAÑA

Apartado 98
Torrejon de Ardoz - Madrid
Tel. 675 53 00
675 53 50

FRANCE

B.P. 28
06021 - Nice Cedex

ITALIA

Casella Postale 1
02015 - Cittaducale
Tel. (0746) 690 34/35/36

NEDERLAND

Postbus 43
Kolthofsingel 8
Almelo
Tel. (05490) 63967

NORGE

Ryensvingen 15
Oslo 6
Tel. (02) 68 94 87

ÖSTERREICH

Marxergasse 10
1030 - Wien
Tel. (0222) 72 41 86

PORUGAL

2650 Rua Engº,
Frederico Ulrich
Moreira Da Maia
Douro
Tel. (02) 948 1003.

SCHWEIZ - SUISSE

Aargauerstraße 250
8048 - Zürich
Tel. (01) 64 34 55/56

SUOMI FINLAND

P.L. 917
Freesenkatu 6
00101 - Helsinki 10
Tel. (80) 40 83 00

SVERIGE

Norra Hamnvägen 3
Fack
100 54 Stockholm 39
Tel. (08) 23 54 80

© 2010 Joerg Woerner

Datamath Calculator Museum

TEXAS INSTRUMENTS