

HEWLETT-PACKARD

HP-65

Bedienungs-Handbuch

INHALTSVERZEICHNIS

EINFÜHRUNG

Die Verwendungsmöglichkeiten des HP-65

1. Manuelle Durchführung

2. Benutzung eines auf dem HP-65 basierenden Programms

3. **HEWLETT  PACKARD**

4. ALLGEMEINE BETRIEBUNGSANWEISUNGEN

Einrichten des HP-65

Anzeige

Programmspeicher und Memory-Link

Case A-Betrieb

Erweiterung und Verbindung des Stack-Links

5. SPEICHERREGISTER

Speicherregister

Zielfunktion

6. FUNKTIONEN

Bedienungs-Handbuch

7. PROGRAMMIERUNG

Wie man programmiert

Die HP-65-Programme

INHALTSVERZEICHNIS

EINLEITUNG

Drei Verwendungsmöglichkeiten des HP-65	3
1. Manuelle Durchführung von Rechnungen	5
2. Benutzung eines auf Magnetkarte aufgezeichneten Programms	10
3. Erstellen eigener Programme	13

1. ALLGEMEINE BEDIENUNGSANWEISUNGEN

Löschfunktionen	18
Anzeige	19
Eingabe großer und kleiner Zahlen	24
Last X-Register	25
Ermöglichung und Verhinderung des Stack-Lifts.	27

2. SPEICHERREGISTER

Adressierbare Register	29
Zusätzliche Stack-Operationen	33
Aufrufen der Zahl π	36
Eine Zinseszinstabelle	36

3. FUNKTIONEN

Funktionen, die sich auf Winkel beziehen	39
Umwandlungen	45
Funktionen von x und die allgemeine Exponentialfunktion (y^x)	47

4. PROGRAMMIERUNG

Was ist ein Programm?	52
Der Programmspeicher	54

2 Inhaltsverzeichnis

Korrekturoperationen	70
Programmierte Entscheidungen	83
Weitere Bemerkungen zur Programmierung	101
Endgültige Programmkorrektur	106

ANHANG

Anhang A: Betriebsgrenzen	111
Anhang B: Zubehör	114
Anhang C: Pflege und Wartung	116
Anhang D: Typische Fehler	127

STICHWORTVERZEICHNIS	129
----------------------------	-----

ÜBERSICHT ÜBER DIE TABELLEN

Benutzungsanweisungen für das Zinseszins-Programm ...	13
Fehler, die zum Blinken der Anzeige führen	23
Funktionen, die sich auf Winkel beziehen	40
Umwandlungen von x	46
Funktionen von x und die allgemeine Exponentialfunktion (y^x)	48
Speicher, Codes und die Einzelschritt-Taste	57
Korrekturoperationen	68
Operationen für programmierte Entscheidungen	83
Graphische Darstellung der einzelnen Programmaufgaben	107

EINLEITUNG

DREI VERWENDUNGSMÖGLICHKEITEN DES HP-65

Wir beglückwünschen Sie zum Erwerb Ihres programmierbaren Taschenrechners. Zusätzlich zu all den Rechenfähigkeiten, die die früheren Modelle HP-35 und HP-45 so populär gemacht haben, bietet Ihnen Ihr neuer HP-65 eine Möglichkeit, über die kein anderer Taschenrechner verfügt: echte Programmierbarkeit.

Vereinfacht ausgedrückt versteht man unter der Programmierbarkeit des HP-65 seine Fähigkeit, die zur Lösung eines speziellen Problems erforderliche Tastenfolge zu lernen, sich daran zu erinnern und sie automatisch auszuführen. Wie wertvoll diese Eigenschaft ist, erkennt man an der Tatsache, daß die meisten von uns, soweit Sie sich regelmäßig mit Zahlen beschäftigen, einen Großteil Ihrer Zeit auf die Durchführung immer wiederkehrender Rechenschritte verwenden. Ob wir nun Flugpläne ausarbeiten, Vermessungsaufgaben bearbeiten, Kapitalerträge berechnen oder Stromversorgungsteile entwickeln, stets sind diese Aufgaben mit sich wiederholenden, zeitaufwendigen Rechenproblemen verbunden, die unsere Produktivität einschränken und das Erreichen unserer Ziele erschweren. Obwohl es bereits seit einiger Zeit programmierbare Computer und Tischrechner gibt, sind sie für viele Aufgaben ungeeignet, da sie zu teuer, kompliziert und ortsgebunden sind. Die wahre Bedeutung des HP-65 liegt darin, daß er diese Schranken durchbricht und die Vorzüge programmierbarer Rechengeräte – Schnelligkeit, Genauigkeit und Komfort – nahezu jedem bietet. Sie können dieses leistungsfähige Gerät auf drei Arten benutzen:

1. ZUR MANUELLEN DURCHFÜHRUNG VON RECHNUNGEN

Sie verfolgen jeden einzelnen Schritt der Rechnung, indem Sie die Tasten in der tatsächlichen Reihenfolge der Operationen

4 Einleitung

drücken: Sie geben Werte ein, führen Funktionen aus, speichern Ergebnisse, steuern das Anzeigeformat usw., indem Sie die jeweiligen Tasten drücken.

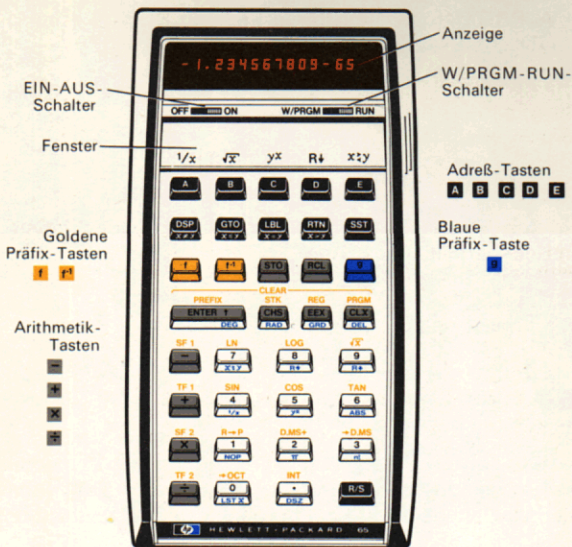
2. ZUR BENUTZUNG EINES AUF MAGNETKARTE AUFGEZEICHNETEN PROGRAMMS

Wenn Sie beschriebene Magnetkarten verwenden (solche gehören zum Beispiel zu dem Standard-Paket, das Ihnen mit Ihrem Rechner geliefert wurde), können Sie mit geringem Aufwand und wenigen Kenntnissen über das Gerät selbst sehr anspruchsvolle Rechnungen ausführen. Sie lesen einfach eine Karte in den Rechner ein und überlassen dem Programm die Ausführung der Berechnungen. In der Regel geben Sie lediglich die Eingangsdaten ein und starten das Programm. Das Programm hält an, wenn weitere Werte benötigt werden oder ein Ergebnis angezeigt werden soll.

3. ZUM ERSTELLEN, SPEICHERN UND AUSFÜHREN IHRER EIGENEN PROGRAMME

Die Programmierung des HP-65 erfordert keine Erfahrungen im Erstellen von Programmen. Sie können den Tasten in der oberen Reihe auf einfache Weise selbstgeschriebene Funktionen zuordnen, die sie für sich allein oder im Zusammenhang mit anderen Programmen verwenden können. Sie planen Ihre Aufgabe über die Tastenfolge, die zur Lösung Ihres Problems nötig ist und die zusätzlichen Tasten, die zur Kontrolle des Programms verwendet werden. Sie schalten den Rechner in den Programm-Modus (Schalterstellung W/PRGM) und tasten die Folge der Einzelschritte in den Programmspeicher ein. Sie können Ihr Programm für die spätere Wiederverwendung aufzeichnen, indem Sie einfach eine Magnetkarte durch den Rechner laufen lassen. Wenn Sie jetzt wieder in den RUN-Modus zurückschalten, können Sie Ihr gespeichertes Programm ausführen.

Im Rahmen dieser Einführung wollen wir diese drei Methoden kurz vorstellen. Wir empfehlen Ihnen, die Beispiele auszuführen.



Sie können sich so über die einwandfreie Funktion Ihres Rechners vergewissern und werden mit ihm vertraut.

1. MANUELLE DURCHFÜHRUNG VON RECHNUNGEN

ZU BEGINN

Ihr HP-65 wird komplett mit eingesetzter aufladbarer Batterie geliefert. Laden Sie die Batterie 14 Stunden lang wie in Anhang C beschrieben, bevor Sie den Rechner netzunabhängig verwenden. Sie können den Rechner aus der Batterie speisen oder ihn an das Ladegerät anschließen und verwenden, während die Batterie geladen wird. Zu Beginn:

6 Einleitung

- W/PRGM-RUN-Schalter in Stellung RUN
- EIN-/AUS-Schalter in Stellung EIN

Der Rechner sollte jetzt **0.00** anzeigen; falls nicht, schlagen Sie im Anhang C nach.

EINGABE VON ZAHLEN

Tasten Sie die Ziffernfolge ein, ebenso den Dezimalpunkt, wenn er Bestandteil der Zahl ist. Versuchen Sie als Beispiel die Zahl 314.32 einzutasten. Sie müssen dazu folgende Tasten drücken:

3 1 4 . 3 2

Unterläuft Ihnen bei der Eingabe einer Zahl ein Fehler, drücken Sie **CLX** (Clear X) um die falsch eingegebene Zahl zu löschen und tasten Sie dann die korrekte Zahl ein.

Negative Zahlen. Tasten Sie zur Eingabe negativer Zahlen zuerst die positive Zahl ein und drücken Sie dann **CHS** (change sign, Vorzeichenwechsel). Um beispielsweise -12 einzugeben:

Drücken Sie 12 **CHS**

Zum Ändern des Vorzeichens einer negativen oder positiven Zahl, drücken Sie **CHS**. Ändern Sie als Beispiel das Vorzeichen der vorangegangenen Zahl, um wieder die positive Zahl 12 zu erhalten:

Drücken Sie **CHS**

ARITHMETISCHE GRUNDRECHNUNGEN

Arithmetische Grundrechnungen werden beim HP-65 durch Drücken einer der Tasten **+**, **-**, **×**, **÷** durchgeführt. Für alle solchen Aufgaben, die aus zwei Zahlen und einem arithmetischen Operator bestehen, tasten Sie die erste Zahl ein und speichern Sie sie durch Drücken der **ENTER**-Taste; geben Sie dann die zweite Zahl ein und drücken Sie die entsprechende Operationstaste. Wollen Sie zum Beispiel 12 und 3 addieren, so drücken Sie:

12 **ENTER** 3 **+**

Der Rechner verwendet die letzte mit **ENTER** gespeicherte Zahl und die zuletzt eingetastete Zahl: er addiert die zuletzt eingetastete Zahl zu der gespeicherten Zahl; er subtrahiert die zuletzt eingetastete Zahl von der gespeicherten Zahl; er multipliziert die gespeicherte Zahl mit der zuletzt eingetasteten Zahl; er dividiert die gespeicherte Zahl durch die zuletzt eingetastete Zahl. Um beispielsweise 3 von 12 zu subtrahieren, drücken Sie:

12 **ENTER** 3 **-**

Um 12 durch 3 zu dividieren, drücken Sie:

12 **ENTER** 3 **÷**

NICHTARITHMETISCHE FUNKTIONEN

Ein blaues Symbol auf der abgeschrägten Tastenvorderseite bezeichnet die Funktion, die ausgeführt wird, wenn diese Taste im Anschluß an die Präfix-Taste **■** gedrückt wird. Ein goldfarbenes Symbol über der Taste bezeichnet die Funktion, die ausgeführt wird, wenn diese Taste im Anschluß an die Präfix-Taste **f** gedrückt wird; die zu dem gleichen goldfarbenen Symbol inverse oder komplementäre Funktion wird ausgeführt, wenn erst **f1** als Präfix und dann die Funktionstaste gedrückt wird. Um eine blaue oder goldfarbene Funktion zu benutzen, drücken Sie die entsprechende Präfix-Taste (**■**, **f** oder **f1**) und im unmittelbaren Anschluß daran die gewünschte Funktionstaste. Beispiel:

Berechnen Sie	Drücken Sie	Anzeige
$\sin(90^\circ) = 1$	90 f SIN	1.00
$\arcsin(.5) = 30^\circ$.5 f1 SIN	30.00
$1/5 = .2$	5 ■ 1/x	0.20

DER ARBEITSREGISTER-STAPEL «STACK»

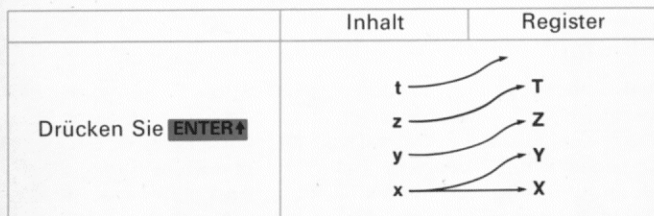
Der HP-65 verfügt über vier mit **X**, **Y**, **Z** und **T** bezeichnete Arbeitsregister. Sie sind in Form eines Stapels angeordnet, wobei das untere Register das **X**-Register ist (siehe unten).

8 Einleitung

Inhalt	Register
t	T
z	Z
y	Y
x	X

Um Verwechslungen zwischen den Bezeichnungen der Register und ihrem Inhalt zu vermeiden, werden die Register selbst in diesem Handbuch mit Großbuchstaben, ihre Inhalte dagegen mit Kleinbuchstaben bezeichnet. So

sind x, y, z und t die jeweiligen Inhalte der Register X, Y, Z und T. Wenn Sie eine Zahl eingeben, so wird diese in das X-Register gespeichert. X ist das Anzeigeregister. Wenn Sie **ENTER** drücken, wird diese Zahl auch in das Y-Register kopiert. Gleichzeitig wird y nach Z und z nach T geschoben. t, der Inhalt des T-Registers, geht verloren (siehe unten):

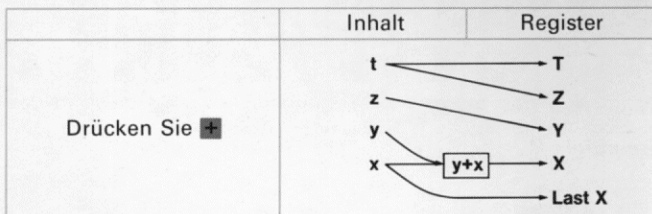


Der HP-65 kann eine Zahl in jedes der vier Register speichern.

Fast alle Aufgaben können gelöst werden, indem die Zahlen in der gleichen Reihenfolge eingetastet werden, in der sie auch in dem ursprünglichen Ausdruck auftreten, das heißt, von links nach rechts. Tasten Sie zur Lösung einer Aufgabe die erste Zahl ein. Falls Sie jetzt eine Operation ausführen können, tun Sie es. Falls nicht, drücken Sie **ENTER**. Geben Sie jetzt die nächste Zahl ein. Führen Sie die Rechenoperation aus, die möglich ist. Gibt es keine Operation, die Sie ausführen können, speichern Sie diese Zahl mit **ENTER** und wiederholen Sie das Verfahren, indem Sie die nächste Zahl eintasten. Die folgenden Beispiele veranschaulichen dieses Verfahren.

Arbeitsweise des «Stack» bei arithmetischen Operationen. Wenn Sie die Additionstaste drücken, wird der Inhalt des X-Registers zu dem Inhalt von Y addiert. Anschließend

wird der «Stack» nach unten verschoben, wobei **t** nach **T** und **Z** geschrieben, **z** nach **Y** geschoben und **(y+x)** nach **X** geschoben wird. Der Inhalt des **X**-Registers wird nach «Last X» abgespeichert (Last X wird in Abschnitt 1 beschrieben).



Dieser Vorgang der Verschiebung des «Stack» findet im Zusammenhang mit sämtlichen arithmetischen Operatoren (+, −, ×, ÷) statt; das jeweilige Resultat erscheint im Anzeigeregister **X**.

Kombinierte arithmetische Operationen. Der HP-65 führt immer dann, wenn im Anschluß an eine Rechenoperation eine neue Zahl eingegeben wird, einen automatischen **ENTER↓**-Befehl aus, der sich auf das Ergebnis der letzten Operation bezieht. Diese Eigenschaft ermöglicht die Durchführung von Serienrechnungen sowie Ketten- und gemischten Kettenrechnungen. Beachten Sie, daß im folgenden Beispiel davon Gebrauch gemacht wird:

Beispiel: $[(4 \times 5) \div (2 + 3)] - 6 = -2$

Drücken Sie

4 **ENTER↓**

5 **×**

2 **ENTER↓**

3 **+**

÷

6 **−**

Anzeige

4.00

20.00

2.00

5.00

4.00

−2.00

Beachten Sie, daß die Zahlen in der Reihenfolge eingegeben werden, in der sie auch in dem zu berechnenden Ausdruck vorkommen. Die Inhalte der Stack-Register zu dem Rechenbeispiel sind in der nachfolgenden Tabelle angeführt.

10 Einleitung

Stack-Register

T												
Z						20	20					
Y		4	4		20	2	2	20		4		
X	4	4	5	20	2	2	3	5	4	6	-2	

Tasten: 4 ↑ 5 × 2 ↑ 3 + ÷ 6 -

Beachten Sie: **ENTER↑** ist hier abgekürzt als **↑**.

Beispiel: $(12 \times 5) + (11 \times 4) + (10 \times 3) = ?$

Drücken Sie

12 **ENTER↑** 5 **×**
 11 **ENTER↑** 4 **×** **+**
 10 **ENTER↑** 3 **×** **+**

Anzeige

60.00
 104.00
 134.00

ERWEITERTE MÖGLICHKEITEN

Der Rechner verfügt neben den Stack-Registern über neun adressierbare Speicherregister, die Zwischenwerte oder häufig benötigte Konstanten speichern können. Damit sind Sie in der Lage, bereits recht komplexe Berechnungen auszuführen, ohne daß Sie Zwischenergebnisse erneut eintasten müssen. Sie haben bis hierher schon einige Erfahrungen im manuellen Rechnen gesammelt. Auf die angesprochenen Speicherregister und weiterreichende Möglichkeiten zur manuellen Durchführung von Rechnungen werden wir im Hauptteil dieses Handbuches eingehen. In der Zwischenzeit wollen wir uns der Benutzung der Programme, die auf Magnetkarte aufgezeichnet sind, zuwenden.

2. DIE BENUTZUNG EINES AUF MAGNETKARTE AUFGEZEICHNETEN PROGRAMMS

Eine im Rechner eingebaute Magnetkarten-Lese/Schreib-Station ermöglicht es, Programme für eine spätere Verwendung auf Programmkarten bereitzuhalten. Verfügt Ihr HP-65 schon so über eine Fülle von Rechenmöglichkeiten, so kann er sich

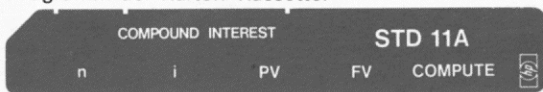
durch das Einlesen solcher Programmkarten zusätzlich innerhalb von Sekunden hochspezialisierte Fähigkeiten aneignen. Durch das Angebot an bereits beschriebenen Magnetkarten für viele Anwendungsbereiche kann diese Möglichkeit auch von solchen Benutzern in Anspruch genommen werden, die sich nicht selbst mit der Programmierung des Rechners befassen möchten.

Sicherlich werden Sie unabhängig von Ihren Hauptinteressen für das Zinseszins*-Programm Verwendung finden. Es ist das 11. Programm des Standard-Paketes, das Ihnen zusammen mit Ihrem Rechner geliefert wurde. Die bereits beschriebene Magnetkarte für dieses Programm finden Sie in der Karten-Kassette zusammen mit 18 weiteren Programmen, einer Reinigungskarte für den Magnetkopf und 20 unbeschriebenen Magnetkarten zur Aufnahme Ihrer eigenen Programme.

Das Standard-Paket setzt sich sowohl aus allgemeinen als auch aus speziellen Programmen zusammen. Einige wurden den verschiedenen Anwendungspaketen entnommen, die von HP bezogen werden können. So zum Beispiel das Programm «Impedanzanpassung eines Pi-Gliedes» aus dem Elektrotechnik-Paket oder «Mittelwert und Standardabweichung» aus dem Statistik-Paket. Wenn es Ihre Zeit erlaubt, können Sie sich mit allen diesen Programmen vertraut machen und die Zahlenbeispiele durchrechnen. Das Zinseszins-Programm ist keinem anderen Programmpaket entnommen. Es wurde für Sie geschrieben, um Ihnen zu ermöglichen, das Kapitalwachstum bei regelmäßigen Spareinlagen zu berechnen. Weitere Angaben zu diesem und den übrigen Programmen des Paketes finden Sie in dem Anweisungsbuch zum Standard-Paket. Im Augenblick wollen wir das Programm einfach einlesen und mit Zahlenbeispielen ausführen.

EINLESEN DES PROGRAMMS

1. Entnehmen Sie die Magnetkarte mit dem Zinseszins-Programm der Karten-Kassette.



* Compound interest STD 11A

12 Einleitung

2. W/PRGM-RUN-Schalter in Stellung RUN.
3. Führen Sie – wie gezeigt – die Karte in den unteren Schlitz an der rechten Gehäuseseite ein. Ist die Karte ein Stück weit eingeschoben, läuft der Motor an und zieht die Karte durch den Rechner zur linken Seite hindurch. Vermeiden Sie es, die Karte beim Transport zu hemmen.



4. Wurde die Karte fehlerhaft gelesen, blinkt die Anzeige und der Programmspeicher wird gelöscht. Drücken Sie **R/S** und lesen Sie die Karte erneut ein.
5. Schieben Sie zum Abschluß die Programmkarte in den oberen Fensterschlitz um die Tasten in der oberen Reihe zu kennzeichnen.

Das Programm steht jetzt im Programmspeicher und kann benutzt werden:

Beispiel: Welcher Betrag ist heute anzulegen, wenn er nach Ablauf von 20 Jahren bei einem quartalsweise zuzurechnenden

Zins bei einem Zinssatz von 7% (pro Jahr) auf DM 15 000,— anwachsen soll?

Folgen Sie zur Lösung der Aufgabe einfach den Anweisungen, die in dem Standardformat in Tabelle 0-1 gegeben sind. Lesen Sie die «Anweisungen» Zeile für Zeile; geben Sie die erforderlichen «Eingabe»-Daten ein, drücken Sie die angegebene(n) Taste(n) und beachten Sie die angezeigten «Ausgabe»-Daten.

Nr.	Anweisungen	Eingabe Werte/ Einheiten	Tasten		Ausgabe Werte/ Einheiten
1	Programm einlesen (Siehe Seite 8)		<input type="text"/>	<input type="text"/>	
2	Programm starten		RTN	R/S	0.00
3	Gebe ein		<input type="text"/>	<input type="text"/>	
	$n(n = 20 \times 4)$	80	A	<input type="text"/>	80.00
	und $i (i = 7 \div 4)$	1.75	B	<input type="text"/>	1.75
	und FV	15000	D	<input type="text"/>	15000.00
4	Berechne PV		E	C	3744.02

Tabelle 0-1. Benützungsanweisungen für das Zinseszins-Programm.

Mit Ihrem HP-65 haben Sie einen Block mit solchen Vor- drucken erhalten. Sie können ihn zusammen mit den 20 un- beschriebenen Magnetkarten verwenden, wenn Sie Programme selbst erstellen.

3. ERSTELLEN EIGENER PROGRAMME

PROGRAMMIERBARKEIT UND DEFINIERBARE TASTEN

Selbst äußerst komplizierte Rechengänge können als Folge einzelner Tastenbefehle dargestellt werden. Da der Rechner voll programmierbar ist, eingeschlossen sind bedingte und unbe- dingte Programmverzweigungen, ist es möglich, Programme

14 Einleitung

aufzustellen, die die ganze Nacht über laufen. Für die Speicherung der Programme im Rechner stehen 100 Speicherpositionen zur Verfügung.

Wir haben bereits gesehen, wie den Tasten in der oberen Reihe des Tastenfeldes spezielle Funktionen zugeordnet werden können, indem eine entsprechende Magnetkarte eingelesen wird. Im Rahmen eines sehr einfachen Beispiels wollen wir jetzt die **A**-Taste definieren. Als erstes planen wir die Funktion, tasten sie dann in den Programmspeicher ein und testen sie anschließend. Sind wir mit den Tests zufrieden, wollen wir das Programm zur späteren Verwendung auf eine Magnetkarte aufzeichnen.

PLANEN DER FUNKTION

Die nachstehende Tastenfolge berechnet x^3 (die dritte Potenz des beliebigen Wertes k , der im X-Register steht).


T				
Z		k		
Y	k	k	k	
X	k	k	k^2	k^3



Taste:    

Beachten Sie: **ENTER**  ist hier abgekürzt als .

Um aus dieser Tastenfolge eine Funktion zu machen, die über die **A**-Taste aufgerufen werden kann, beginnen wir sie mit **LBL A** (zur Kennzeichnung der Funktion) und beenden sie mit **RTN** (womit nach Ablauf der Funktion die Kontrolle an das Tastenfeld zurückgegeben wird).

EINTASTEN DER FUNKTION IN DEN PROGRAMMSPEICHER

1. W/PRGM-RUN-Schalter in Stellung W/PRGM, drücken Sie  **PRGM** zum Löschen des Speichers.
2. Drücken Sie die Tasten in der angegebenen Reihenfolge:

Taste(n)	Bemerkung
LBL	
A	bezeichnet Anfang der Funktion A
ENTER 	
ENTER 	
x	berechnet x^2
x	berechnet x^3
RTN	bezeichnet Ende der Funktion A

Ist Ihnen beim Eintasten der Funktion ein Fehler unterlaufen, löschen Sie (jetzt noch) den gesamten Programmspeicher und beginnen Sie von vorne. Wir werden später noch auf die verschiedenen komfortablen Korrekturmöglichkeiten eingehen und auch die Bedeutung der in der Anzeige erscheinenden Zahlen kennenlernen. Der Rechner hat jetzt «gelernt», auf das Drücken der Taste **A** (im RUN-Modus) die Funktion x^3 auszuführen.

TESTEN DER FUNKTION

- W/PRGM-RUN-Schalter in Stellung RUN.
- Tasten Sie eine Zahl ein und drücken Sie **A**. Der Rechner zeigt jetzt die dritte Potenz der eingegebenen Zahl an. Falls nicht, haben Sie bei der Erstellung der Funktion einen Fehler gemacht.

Drücken Sie	Anzeige	Bemerkung
2 A	8.00	2^3
3 A	27.00	3^3
4 A	64.00	4^3
5 CHS A	-125.00	$(-5)^3$

AUFZEICHNEN DER FUNKTION AUF PROGRAMMKARTE

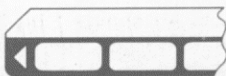
Zur Aufzeichnung des Programms

1. Wählen Sie eine ungeschützte Magnetkarte aus (der Eckenabschnitt darf **nicht** entfernt sein).



◀ ungeschützt

geschützt ▶



16 Einleitung

2. W/PRGM-RUN-Schalter in Stellung W/PRGM (W/PRGM steht für **Write** = Schreiben).
3. Lassen Sie eine Magnetkarte genauso wie beim Einlesen eines Programms durch den Rechner laufen. Die Magnetkarte enthält jetzt Ihr Programm, es sei denn, sie war geschützt. Im obigen Beispiel blieben die Tasten **B** ... **E** undefiniert; selbstverständlich hätten wir auch sie mit Funktionen belegen können.

Wir haben Ihnen jetzt gezeigt, wie Sie das Programm zu einer einfachen Funktion erstellen und es einer der fünf Tasten zuordnen können, die durch den Benutzer definiert werden. Ihr HP-65 kann aber auch programmiert werden, ohne daß auf die Tasten in der oberen Reihe Bezug genommen wird. Ein Beispiel dazu wird Ihnen an späterer Stelle begegnen, wenn wir uns noch einmal dem Thema der Programmierung zuwenden. Sie werden dann sehen, wie mit wenig zusätzlichem Aufwand Funktionen und andere Programme erstellt werden können, deren Leistungsfähigkeit weit über das hier vorgestellte Primitiv-Beispiel hinausgeht.

VORSCHAU

Wir hoffen, daß Sie die Einführung in den Umgang mit Ihrem programmierbaren Taschenrechner HP-65 mit Interesse gelesen haben. Der übrige Teil des Handbuches geht auf all die Eigenschaften des Rechners ein, die bisher noch nicht vorgestellt wurden. Sicherlich genügt es, wenn Sie dieses Material erst einmal oberflächlich überfliegen und dabei auf die eingerahmten Teile achten. Wenn Sie auf diese Weise das ganze Handbuch durchgegangen sind, können Sie sich bei einem gründlicheren Lesen den Details zuwenden. Sie werden sehr schnell mit dem Tastenfeld des Rechners vertraut werden und Sicherheit im Umgang mit dem Gerät gewinnen, wenn Sie die zahlreichen Beispiele durchrechnen und sich überzeugen, daß Sie sie verstanden haben.

Die Abschnitte **1** bis **3** beziehen sich auf die manuelle Durchführung von Rechnungen. Abschnitt **1** erklärt die Wahl des Anzeigeformates und die Eingabe von Daten in wissenschaft-

licher Schreibweise. Abschnitt 2 erklärt den Gebrauch der adressierbaren Speicherregister und die Handhabung des Stack, während im Abschnitt 3 die Benutzung der vorprogrammierten Funktionen besprochen wird.

Der Abschnitt 4 befaßt sich mit der Programmierung. Es wird Ihnen gezeigt, wie Sie Ihre Programme berichtigen und abändern können, wie man ein Programm veranlaßt, Teilbereiche zu wiederholen, an bestimmten Stellen anzuhalten, Entscheidungen zu treffen und vieles mehr. Dieser Abschnitt veranschaulicht Ihnen auch den Gebrauch der Arbeitsblätter für die Programmierung, die mit Ihrem Rechner geliefert wurden.

DIE HP-65-KURZANLEITUNG

In dieser Kurzanleitung sind die wesentlichen Bedienungsverfahren zusammengefaßt und die Funktion der einzelnen Tastenbefehle erklärt. Zum leichteren Nachschlagen der Details sind die Erläuterungen zu den Tasten in der alphabetischen Reihenfolge der Symbole gegliedert. Die HP-65-Kurzanleitung kann mit dem Rechner zusammen in der Weichledertasche mitgeführt werden.

ALLGEMEINE BEDIENUNGSANWEISUNGEN

In diesem Abschnitt wird beschrieben • wie die verschiedenen Speicher gelöscht werden • wie das Anzeigeformat eingestellt wird • wie Zahlen in wissenschaftlicher Schreibweise eingegeben werden • wie mit Hilfe des Last X-Registers Fehler beseitigt werden können, wenn Sie falsche Tasten gedrückt haben. Außerdem werden am Ende des Abschnitts in einem eingerahmten Teil Angaben zum «Stack Lift» (automatische Eingabe) zusammengefaßt.

LÖSCHFUNKTIONEN

Der HP-65 verfügt über vier verschiedene Löschoperationen. Um sie auszuführen, werden die **f**-Funktionen der Tasten in der vierten Zeile des Tastenfeldes benutzt.

LÖSCHEN EINES NICHT GEWÜNSCHTEN PRÄFIX

f **PREFIX** hebt die Wirkung einer fälschlicherweise gedrückten Präfixtaste auf, so daß Operationen ausgeführt werden können, für die keine Präfixtaste zu drücken ist. Nehmen wir an, Sie haben versehentlich **f**, **f1** oder **g** gedrückt, bevor Sie eine Zahl eintasten wollen. Wenn Sie jetzt die Zifferntaste drücken, werden Sie die entsprechend andere Funktion dieser Taste ausführen, anstatt eine Ziffer einzugeben. Um dies zu vermeiden drücken Sie **f** **PREFIX** zur Aufhebung des unerwünschten Präfix und tasten dann, wie beabsichtigt, die Zahl ein. Wurde eine falsche Präfixtaste gedrückt und ist ein anderer Präfix erwünscht, so genügt es, die entsprechend richtige Präfixtaste zu drücken und von da fortzufahren.

Die folgenden (noch nicht erläuterten) Tasten sind ebenfalls Präfixtasten:

STO **RCL** **DSP** **GTO** **LBL**

LÖSCHEN DER STACK-REGISTER

f **STK** löscht alle vier Register (X, Y, Z und T) des Arbeitsregister-Stapels (Stack). Soll nur das X-Register gelöscht werden, drücken Sie **CLX**.

LÖSCHEN DER ADRESSIERBAREN REGISTER

f **REG** löscht alle neun adressierbaren Speicherregister. (Die Register werden in einem späteren Abschnitt besprochen.)

LÖSCHEN DES PROGRAMMSPEICHERS

f **PRGM** löscht die 100 Zeilen des HP-65-Programmspeichers, falls der W/PRGM-RUN-Schalter in Stellung W/PRGM steht. Im RUN-Modus wirkt **f** **PRGM** genauso wie **CLX**.

LÖSCHEN DES GESAMTEN RECHNERS

Sämtliche Speicherregister (einschließlich Programmspeicher) des Rechners können Sie löschen, indem Sie das Gerät erst aus- und dann wieder einschalten. Beachten Sie dabei, daß beim Einschalten des Rechners automatisch die fünf Unterprogramme in den Programmspeicher geladen werden, die zu den Symbolen im Fenster über den Tasten in der oberen Reihe gehören ($\frac{1}{x}$, \sqrt{x} , y^x , $R\downarrow$, $x\div y$).

ANZEIGE


Die Anzeige hat mehrere Funktionen. Sie wird benutzt zum Anzeigen der Ergebnisse, für Fehlermeldungen, zur Anzeige abfallender Batteriespannung, zur Mitteilung, daß der Rechner bei der Ausführung eines Programms ist und zur Anzeige einzelner

20 Allgemeine Bedienungsanweisungen

Programmschritte. Im W/PRGM-Modus ermöglicht Ihnen die Anzeige zusätzlich, die einzelnen Programmbefehle im Programmspeicher darzustellen (diese Verwendung der Anzeige wird im Abschnitt Programmierung besprochen).

EINSTELLEN DES ANZEIGEFORMATES

Der HP-65 kann bis zu 15 Zeichen darstellen: Vorzeichen der Mantisse, zehnstellige Mantisse, Dezimalpunkt, Vorzeichen des Exponenten und zweistelligen Exponenten. Im RUN-Modus wird der abgerundete Inhalt des X-Registers angezeigt. Vom Tastenfeld aus können zwei Anzeige-Modi (Festkomma- oder wissenschaftliche Darstellung) mit einer Vielfalt von Rundungsmöglichkeiten gewählt werden. (Die verschiedenen Rundungsweisen beziehen sich nur auf die Anzeige; stets bleibt die volle Genauigkeit im HP-65 intern erhalten.)

Festkomma-Anzeige. Die Festkomma-Darstellung wählen Sie durch Drücken von **DSP** , gefolgt von einer entsprechenden Zifferntaste zur Angabe der gewünschten Zahl von Nachkommastellen (0–9). Die Festkomma-Darstellung ermöglicht es, alle Ergebnisse mit gleicher Genauigkeit anzuzeigen. Die Anzeige ist links bündig und zeigt im Rahmen des gewählten Formates auch Nullen hinter dem Komma an. Wird der Rechner aus- und dann eingeschaltet, wählt der Rechner selbstständig Festkomma-Darstellung mit zwei Nachkommastellen als Anzeigeformat. Zum Beispiel:

Drücken Sie

(Achten Sie darauf, daß der W/PRGM-RUN-Schalter in Stellung RUN steht. Schalten Sie den Rechner erst aus, dann ein.)

123.4567 **ENTER**

DSP  **4**

DSP  **6**

DSP  **2**

DSP  **0**

Anzeige

0.00

123.46

123.4567

123.456700

123.46

123.

Wissenschaftliche Schreibweise. Sie eignet sich für das Arbeiten mit großen oder sehr kleinen Zahlen und ermöglicht die Anzeige der Ergebnisse mit der stets gleichen Zahl wesentlicher Stellen. Sie wählen dieses Anzeigeformat durch Drücken von **DSP**, gefolgt von einer Zifferntaste zur Angabe der gewünschten Zahl von Nachkommastellen der Mantisse. Auch hier wird lediglich die Anzeige gerundet; sie ist links bündig und beinhaltet Nachkomma-Nullen innerhalb des gewählten Formates. Zum Beispiel:

Drücken Sie	Anzeige	Bemerkung
(Schalten Sie den Rechner erst aus, dann ein.)	0.00	
123.4567 ENTER	123.46	
DSP 2	1.23 02	$= 1.23 \times 10^2$
DSP 4	1.2346 02	$= 1.2346 \times 10^2$
DSP 8	1.23456700 02	$= 1.234567 \times 10^2$

Schalten Sie jetzt um auf 8 Dezimalstellen in Festkomma-darstellung.

DSP **□** **8** 1.234567000 02 * $= 1.234567 \times 10^2$

Schalten Sie jetzt zurück auf Festkommadarstellung mit zwei Nachkommastellen.

Drücken Sie	Anzeige
DSP □ 2	123.46
□ 0005 CHS ENTER	-0.00 (*)

BLINKENDE ANZEIGE

Die Anzeige blinkt bei dem Versuch, eine unerlaubte Operation auszuführen. Sobald irgendeine Taste gedrückt wird, hört das Blinken auf, ohne daß die dieser Taste zugeordnete Funk-

* Ist eine Zahl zu groß, um in dem gewählten Format dargestellt zu werden, zeigt sie der Rechner selbstständig in wissenschaftlicher Schreibweise mit 10 Stellen (volle Genauigkeit) an.

* Ist ein Ergebnis zu klein, um im gewählten Format angezeigt werden zu können, wird Null angezeigt (mit einem Minuszeichen, falls das Resultat negativ ist).

22 Allgemeine Bedienungsanweisungen

tion ausgeführt wird. Zweckmäßigerweise wird das Blinken der Anzeige mit **CLX** gestoppt. Die unerlaubten Operationen, die zu einer solchen Fehleranzeige führen, sind in Tabelle 1-1 aufgeführt.

UNLESBARE ANZEIGE

Während der Ausführung eines gespeicherten Programms ändert sich die Anzeige laufend; sie ist absichtlich unlesbar, um das Arbeiten eines Programms anzuzeigen. Sobald das Programm anhält, erscheint wieder eine stehende Anzeige.

ANZEIGE ALLER DEZIMALPUNKTE

Die Kapazität der Batterie erlaubt etwa 3 Stunden ununterbrochenen Betrieb. Sie können dadurch Energie sparen, daß Sie den Rechner immer dann abschalten, wenn Sie ihn nicht gerade benutzen. Auch wenn Sie kein Programm oder Ergebnis verlieren möchten (den Rechner also nicht ausschalten können), ist es möglich, den Stromverbrauch zu senken: tasten Sie ein ☐ ein und belassen Sie den Rechner in dem Zustand, bis Sie ihn wieder benötigen.

Das Aufleuchten aller Dezimalpunkte der Anzeige bedeutet, daß die Batterie bald erschöpft ist und Ihnen nur noch etwa 2 bis 5 Minuten Betriebszeit verbleiben. Auch wenn sämtliche Punkte aufleuchten, ist die Stellung des Dezimalpunktes klar, da ihm eine ganze Anzeigestelle zugeordnet ist.

Beispiel:



↑
wahrer Dezimalpunkt

Falls alle Dezimalpunkte aufleuchten, während der Motor des Kartenlesers läuft und dann ausgehen, ist die Batterie weitgehend entladen.

Tasten	Funktion	Fehler
f LN	Natürlicher Logarithmus (Basis e)	$x \leq 0$
f LOG	Logarithmus zur Basis Zehn	$x \leq 0$
f \sqrt{x}	Quadratwurzel	$x < 0$
f-1 SIN	Arcsin	$ x > 1$
f-1 COS	Arccos	$ x > 1$
f DMS+	{Addiert } Grad, Minuten,	$ x $ oder $ y $ oder
f-1 DMS+	{Subtrahiert } Sekunden	$ x \pm y > 99999.59599$ D.MS
f →DMS	{ wandelt dezimal gegebene Winkel }	$ x > 99999.99999$ Grad oder
f-1 →DMS	{ nach/von Grad, Minuten, }	entsprechend in rad oder Neugrad
f →OCT	{ Sekunden um }	
f →OCT	Dezimal nach oktal	x nicht ganzzahlig oder
f-1 →OCT	Oktal nach dezimal	$ x > 1073741823_{10} = 7777777777_8$
g $1/x$	Reziprokwert	x nicht ganzzahlig oder
g y^x	Exponentialfunktion	$ x > (1222222221)_8 =$
g n!	Fakultät	$9999999999_8 = 1380525201_{10}$
÷	Division	$x = 0$
	Lesen einer Magnetkarte	$y \leq 0$
		x nicht ganzzahlig oder $x < 0$
		$x = 0$
		unbeschriebene Karte oder Daten-
		fehler beim Einlesen

Tabelle 1-1. Fehler, die zum Blinken der Anzeige führen.

24 Allgemeine Bedienungsanweisungen

Das Betreiben des Rechners über die angegebenen 2 bis 5 Minuten ab erstmaligem Aufleuchten der Punkte hinaus kann zu Fehlern in der Anzeige führen. Die Batterie muß ersetzt werden, oder Sie laden sie, indem Sie den Rechner an das Ladegerät anschließen. Achten Sie darauf, daß der Rechner ausgeschaltet ist, wenn Sie die Verbindung zum Ladegerät herstellen.

Achten Sie auch darauf, daß die Batterie zumindest teilweise geladen ist, wenn Sie die Magnetkarten-Lese/Schreib-Einrichtung benutzen wollen.

EINGABE GROSSER UND KLEINER ZAHLEN

Sie können Zahlen mit einem Exponentialfaktor (zur Basis 10) eingeben, indem Sie die Taste **EEX** (Eingabe des Exponenten) drücken. Geben Sie beispielsweise 15.6 Billionen (15.6×10^{12}) ein und multiplizieren Sie sie mit 25.

Drücken Sie	Anzeige	Bemerkung
15.6 EEX	15.6 00	
12	15.6 12	15.6×10^{12} *
ENTER \uparrow	1.560000000 13	1.56×10^{13}
25 \times	3.900000000 14	Ergebnis

GENAUE ZEHNERPOTENZEN

Bei der Eingabe genauer Zehnerpotenzen können Sie Zeit sparen, indem Sie **EEX** drücken und dann die entsprechende Potenz zu Zehn. Geben Sie beispielsweise 1 Million (10^6) ein und dividieren Sie sie durch 52.

Drücken Sie	Anzeige	
EEX 6	1. 06	
ENTER \uparrow	1000000.00	
52 \div	19230.77	

Zur Anzeige des Resultates in wissenschaftlicher Schreibweise mit 6 Dezimalstellen,

* Um eine negative Zahl (d.h. -15.6×10^{12}) einzutasten, drücken Sie **CHS**, dann **EEX**.

Drücken Sie**DSP** **6****Anzeige**

1.923077 04

KLEINE ZAHLEN (NEGATIVER EXPONENT)

Zur Eingabe negativer Exponenten, tasten Sie die Zahl ein, drücken Sie **EEX** und anschließend **CHS**, um den Exponent negativ zu machen und geben Sie dann die entsprechende Zehnerpotenz ein. Tasten Sie als Beispiel die Planck'sche Konstante (h) – ungefähr 6.625×10^{-27} erg sec – ein und multiplizieren Sie sie mit 50.

Drücken Sie6.625 **EEX**

27

CHS**ENTER** **↑**50 **x****Anzeige**

6.625 00

6.625 27

6.625 -27

6.625000 -27

3.312500 -25

Wenn Sie die Anzeige auf **DSP** **•** **2** zurückstellen, wird das Ergebnis als Null angezeigt.

LAST X-REGISTER

Last X ist die Bezeichnung für ein Register, das ausschließlich dazu verwandt wird, den letzten Wert von **x** zu speichern (d. h. die Zahl, die in der Anzeige erscheint), wenn inzwischen eine Operation ausgeführt wurde, die diesen **x**-Wert benutzt hat. Beim Einschalten des Rechners wird der Inhalt von Last X Null gesetzt. Der Registerinhalt ändert sich erst dann, wenn eine Operation ausgeführt wird, die sich auf **x** oder **x** und **y** bezieht; in diesem Fall wird nach Drücken der Operationstaste zuerst **x** nach Last X gespeichert und dann erst die entsprechende Operation ausgeführt. Der gespeicherte Wert kann über die Tastenfolge **0** **LST X** (auf Wunsch auch wiederholt) in das X-Register zurückgerufen werden. Last X ist besonders dann sehr wertvoll, wenn Sie Fehler rückgängig machen wollen, die durch das Drücken einer falschen Operationstaste oder das Rechnen mit einem falschen Eingabeargument entstanden sind.

26 Allgemeine Bedienungsanweisungen

Wenn Sie beispielsweise im Rahmen eines längeren Rechnungsganges 3 von 12 subtrahieren wollten und statt dessen dividiert haben, können Sie den Fehler wie folgt kompensieren:

Drücken Sie	Anzeige	Bemerkung
12 ENTER 3 ÷	4.00	Hoppla! – Sie wollten doch subtrahieren.
9 LST x	3.00	Bringt den letzten Wert vor Durchführung der Division zurück.
x	12.00	Gleicht die ungewollte Division aus; Sie sind wieder da, wo Sie begonnen haben.
9 LST x	3.00	Bringt die zuletzt angezeigte Zahl vor Durchführung der Multiplikation zurück.
–	9.00	Richtige Rechenoperation führt zu dem gewünschten Ergebnis.

Last X kann Ihnen viel Mühe ersparen, wenn Sie im Rahmen einer längeren Rechnung eine Zahl korrigieren müssen, da Sie die Rechnung nicht wieder von vorne beginnen müssen. Dividieren Sie beispielsweise 12 durch 2157, nachdem Sie irrtümlicherweise durch 3157 dividiert haben.

Drücken Sie	Anzeige	Bemerkung
12 ENTER 3.157 ÷	3.80	Sie wollten durch 2157 dividieren anstatt durch 3157 .
9 LST x	3.16	Bringt die zuletzt angezeigte Zahl vor Ausführung der letzten Operation zurück.
x	12.00	Sie haben den Fehler beseitigt.

2.157 \div

5.56

Die richtige Operation führt zum gewünschten Resultat.

Eine weitere, vielleicht sogar wichtigere, Anwendung findet Last X innerhalb von Funktionen, bei denen x mehrfach als Argument vorkommt. Ohne auf Details einzugehen, da Funktionen noch nicht näher behandelt worden sind, wollen wir Ihnen einige solcher Beispiele nennen:

$$\frac{\sin x}{x}, y^x - x, \sin x + \cos^3 x$$

Bei jedem dieser Beispiele wird bei der ersten Operation der Wert x nach Last X gespeichert.

Die folgenden Operationen (einschließlich ihrer Inversen) speichern x nach Last X:

+, **-**, **×**, **÷**, **COS**, **→DMS**, **DMS+**, **INT**, **LN**, **LOG**, **→OCT**, **R→P**, **SIN**, **TAN**, **n!**, **√x**, **1/x**, **y^x**, **ABS**. Beachten Sie, daß sich **CLX** nicht auf Last X bezieht.

ERMÖGLICHUNG UND VERHINDERUNG DES «STACK-LIFT»

Wir haben bereits in der Einleitung gesehen, daß bei der Eingabe einer neuen Zahl das vorangehende Zwischenergebnis automatisch im Stack (Rechenregister-Stapel) angehoben wird und uns so das Drücken der Taste **ENTER** erspart wird. Der gleiche Vorgang spielt sich beim Rückrufen von Daten aus einem der adressierbaren Speicherregister, von Last X oder beim Abruf der permanent gespeicherten Zahl π in das X-Register ab. Sie werden beobachtet haben, daß bestimmte andere Operationen ebenfalls den Stack-Lift (Anheben des Rechenregisterstapels) ermöglichen, während **CLX** und **ENTER** diesen Vorgang verhindern. In der Regel werden Sie gar nicht auf den Lift-Status achten, da er sich für die meisten Rechnungen so natürlich auswirkt. Zum Nachschlagen in Zweifels-

28 Allgemeine Bedienungsanweisungen

fallen sind in der nachstehenden Tabelle alle Tasten aufgeführt, die sich auf den Lift-Status auswirken. Beachten Sie dabei, daß eine Reihe von Tasten keinen Einfluß auf den Stack-Lift haben. Die meisten der in der Tabelle aufgeführten Operationen werden noch in diesem Handbuch vorgestellt.

Operationen, die den Stack-Lift verhindern: **R/S** innerhalb eines Programms, falls das Programm unmittelbar vor einem **R/S**-Befehl eine Zahl in das X-Register einliest; **CLX** und **ENTER↑** zu jeder Zeit.

Operationen, die den Stack-Lift ermöglichen: Sämtliche Zahleneingabe-Tasten: **0** ... **9**, **.**, **EEX**, **π**, aber nicht **CHS**. • Alle Rechentasten: **-**, **+**, **x**, **÷**, **ABS**, **COS**, **→DMS**, **DMS+**, **INT**, **LN**, **LOG**, **LST x**, **nl**, **→OCT**, **R→P**, **SIN**, **TAN**, **1/x**, **√x**, **y^x**. • Tasten zur Beeinflussung des Stack: **R↑**, **R↓**, **x₁↔y**, aber nicht **ENTER↑**. • Speicherregister-Tasten: **STO**, **RCL**.

Operationen, die keinen Einfluß auf den Stack-Lift haben: Sämtliche anderen Tasten haben auf den Stack-Lift keinen Einfluß. Dies umfaßt: alle Programmier Tasten, Winkelmodus-Tasten, Tasten zur Kontrolle der Anzeige, Löschtasten (außer **CLX**), und **CHS**.

ABSCHNITT 2

SPEICHERREGISTER

In diesem Abschnitt werden wir den Gebrauch der neun adressierbaren Speicherregister beschreiben und auf den Umgang mit dem Inhalt des Stack eingehen. Als Beispiel für die Verwendung der Stack-Register wird die manuelle Berechnung eines Zinseszinsproblems vorgestellt.

ADRESSIERBARE REGISTER

Es stehen neun adressierbare Speicherregister R_1, R_2, \dots, R_9 zur Verfügung. Ihre jeweiligen Inhalte wollen wir mit r_1, r_2, \dots, r_9 bezeichnen. Zu den typischen Verwendungen der Speicherregister zählen das Speichern von Konstanten, von laufenden Summen oder Zwischenergebnissen. Sie können den Inhalt des **X**-Registers in ein beliebiges der neun adressierbaren Register abspeichern oder den Inhalt eines dieser Register nach **X** zurückerufen. Zusätzlich können Sie in ein beliebiges dieser Register eine arithmetische Summe, Differenz, Produkt oder einen Quotienten aus dem Inhalt des gegebenen Registers und dem **X**-Register speichern. Wenn zum Beispiel der Inhalt von R_5 gleich 100 ist und **X** den Wert 70 enthält, können Sie die Differenz ($100 - 70 = 30$) nach R_5 speichern, indem Sie einfach drücken: **STO** **-** **5**.

SPEICHERN UND RÜCKRUFEN VON DATEN

Zum **Speichern** einer angezeigten Zahl (egal, ob es das Ergebnis einer Rechnung oder eine eingetastete Zahl ist):

1. Drücken Sie **STO**.
2. Drücken Sie eine der Zifferntasten **1** bis **9** zur Wahl des Registers, in das die Zahl gespeichert werden soll.

Ist in dem gewählten Register schon eine Zahl gespeichert, so

30 Speicherregister

wird sie durch den neuen Wert überschrieben. Der Inhalt von **X** bleibt unverändert.

Zum **Rückrufen** einer Zahl, die in einem der neun Speicherregister steht:

1. Drücken Sie **RCL**.
2. Drücken Sie eine der Zifferntasten (**1** ... **9**) zur Angabe, aus welchem der Register die Zahl zurückzurufen ist.

Durch das Zurückrufen eines Wertes aus einem der Register wird dessen Inhalt nicht zerstört. Es wird lediglich eine Kopie des Inhaltes in das Anzeigeregister übertragen – das Original bleibt so lange im Register stehen, bis entweder: (1) eine neue Zahl in dieses Register gespeichert wird, (2) der Rechner ausgeschaltet wird, oder (3) alle neun Speicherregister mit der Tastenfolge **f** **REG** gelöscht werden. Das Zurückrufen eines Wertes aus einem Speicherregister verursacht ein Anheben des Stacks, es sei denn, der Stack-Lift ist verhindert.

Beispiel 1: Ein Kunde hat drei Artikel zu folgenden Preisen gekauft: DM 1000,—, DM 2000,— und DM 3000,—. Sie sollen nun auf alle Verkäufe über DM 5000,— ein Skonto von 5% gewähren. Wieviel hat also der Kunde für die einzelnen Artikel zu bezahlen und wie hoch ist der Gesamtbetrag?

Lösung:

Drücken Sie	Anzeige	Bemerkung
1 ENTER ↓ .05 — STO 1	0.95	Speichern der Konstante 0,95—95% in R ₁
1000 RCL 1 ×	950.00	Betrag für den ersten Artikel.
2000 RCL 1 ×	1900.00	Betrag für den zweiten Artikel.
3000 RCL 1 ×	2850.00	Betrag für den dritten Artikel.
+ +	5700.00	Gesamtbetrag.

Beispiel 2. Nachstehend ist der Inhalt und die Höhe von drei verschiedenen Tanks angegeben. Sie sollen die US-

Einheiten in die entsprechenden metrischen Einheiten umwandeln und Volumen und Höhe der Tanks in Liter bzw. Zentimeter umrechnen.

	Fassungsvermögen (gal.)	Höhe (in.)
Tank 1	3.6	13.5
Tank 2	5.5	20.9
Tank 3	11.3	32.8

Berücksichtigen Sie folgende Beziehungen:

1 U.S. Gallone = 3,7854 Liter

1 inch (Zoll) = 2,5400 Zentimeter

Diese Konstanten wollen wir nach R_1 und R_2 abspeichern.

Drücken Sie	Anzeige	Bemerkung
DSP <input type="checkbox"/> 4		
3.7854 STO 1	3.7854	Speichert Umwandlungskonstante Liter/gal. nach R_1 .
2.54 STO 2	2.5400	Speichert Umwandlungskonstante cm/in. nach R_2 .
3.6 RCL 1 x	13.6274	Volumen von Tank 1 in Liter.
13.5 RCL 2 x	34.2900	Höhe von Tank 1 in Zentimeter.
5.5 RCL 1 x	20.8197	Volumen von Tank 2 in Liter.
20.9 RCL 2 x	53.0860	Höhe von Tank 2 in Zentimeter.
11.3 RCL 1 x	42.7750	Volumen von Tank 3 in Liter.
32.8 RCL 2 x	83.3120	Höhe von Tank 3 in Zentimeter.
DSP <input type="checkbox"/> 2	83.31	Stellt Anzeige zurück.

AUSWAHL DER ADRESSIERBAREN REGISTER

Mit Ausnahme der Register R_8 und R_9 , ist es Ihnen frei überlassen, welche Register Sie belegen wollen.

32 Speicherregister

R_8 wird von der Operation «Dekrement und Sprung bei Null» (**DSZ**) belegt. Diese Operation, die noch in Abschnitt 4 besprochen werden wird, nutzt R_8 als Zählregister für das Durchlaufen von Programmschleifen. Wenn Sie den Gebrauch dieser Operation beabsichtigen, sollten Sie das Register R_8 nicht andersweitig verwenden. Anderenfalls können Sie auch über R_8 frei verfügen.

R_9 wird vom Rechner beim Aufruf trigonometrischer Funktionen (einschließlich der Umwandlung von rechtwinkligen in Polar-Koordinaten) benutzt, ebenso bei der (in Programmen üblichen) Anwendung der Verhältnis-Tests. Diese Operationen nutzen R_9 für interne Zwischenrechnungen. Zu anderen Zeiten ist auch R_9 frei verfügbar.

Die folgenden Operationen zerstören den Inhalt von R_9 :
SIN, **COS**, **TAN**, **R→P** (trigonometrische Funktionen und ihre Umkehrfunktionen);

x≠y, **x≤y**, **x=y**, **x>y** (Verhältnis-Tests).

RECHNEN IN ADRESSIERBAREN REGISTERN

Bis jetzt haben sich alle Rechenschritte auf das **X**-Register oder **X** und **Y** bezogen, wobei das Resultat jeweils in **X** stand. Bei der Durchführung von Rechnungen in den Speicherregistern wird das Ergebnis selbst in das Register gespeichert und **x** bleibt unverändert.

Subtraktion: Um **x** von r_n zu subtrahieren, drücken Sie **STO -** **[n]**.

Addition: Um **x** zu r_n zu addieren, drücken Sie **STO +** **[n]**.

Multiplikation: Um **x** und r_n miteinander zu multiplizieren, drücken Sie **STO x** **[n]**.

Division: Um r_n durch **x** zu dividieren, drücken Sie **STO ÷** **[n]**.

Speichern Sie beispielsweise 6 in R_1 und addieren Sie dann 2 dazu.

Drücken Sie

 6 **STO** **1**
 2 **STO** **+** **1**
RCL **1**
Anzeige

 6.00
 2.00
 8.00

Bemerkung

 Speichert 6 nach R_1
 Addiert 2 zu r_1
 Bestätigt, daß $r_1 = 8$

 Subtrahieren Sie jetzt 5 von r_1 .

 5 **STO** **-** **1**
RCL **1**

 5.00
 3.00

 Bestätigt, daß r_1 auf 3 verringert wurde.

 Multiplizieren Sie abschließend den verbleibenden Inhalt von R_1 mit 2:

 2 **STO** **x** **1**
RCL **1**

 2.00
 6.00

 Bestätigt, daß r_1 auf 6 erhöht wurde.

ZUSÄTZLICHE STACK-OPERATIONEN

Sie wissen bereits, daß mit Ausnahme der gerade besprochenen Register-Arithmetik sämtliche Rechnungen im Stack ausgeführt werden. Sie sorgen einfach dafür, daß die Zahlenwerte an ihren richtigen Stellen stehen und drücken dann die gewünschte Operationstaste. Von der Möglichkeit, Zwischenergebnisse über den Zahlen im Stack zu speichern, mit denen im Augenblick gerechnet wird, wissen Sie schon. Zum Verschieben der Daten im Stack nach oben haben Sie die **ENTER↑**-Taste und den automatischen Stack-Lift verwendet. Bei der Durchführung der arithmetischen Grundrechnungsarten wurde der Rechen-

R↑	R↓	x↔y
Roll Up – zyklische Vertauschung nach oben	Roll Down – zyklische Vertauschung nach unten	Austausch zwischen x und y

34 Speicherregister

register-Stapel nach unten verschoben. Wir werden jetzt die drei noch verbleibenden Möglichkeiten vorstellen, Daten im Stack zu verschieben und umzuspeichern.

Diese Operationen werden verwendet, wenn Sie die Inhalte der Stack-Register anzeigen wollen (d. h. nach **X** bringen müssen) oder wenn Sie für eine Rechnung Daten in die richtigen Positionen schieben wollen.

Beachten Sie, daß

- **R↑** Ihnen auf einfache Weise ermöglicht, **t** anzuzeigen.
- **R↓** **x** in das **T**-Register abspeichert.
- **xzy** Ihnen ermöglicht, **y** anzuzeigen, ohne daß dabei **z** und **t** verändert werden.
- **xzy**, **R↑**, **R↓** Alternativ-Funktionen zu den Tasten **[7]**, **[8]**, **[9]** sind.

R↓ und **xzy** sind ebenfalls über die **D**- und **E**-Taste verfügbar, wenn der Rechner eingeschaltet wird und der Programmspeicherinhalt inzwischen noch nicht verändert wurde. Die fünf Funktionen, deren Symbole in dem Fenster über der obersten Tastenreihe stehen, wurden ausgewählt, weil sie häufig benötigt werden. Sie sind für die manuelle Durchführung von Rechnungen vom Tastenfeld aus gedacht und können so mit nur einem Tastendruck, anstatt wie sonst mit einem vorangestellten Präfix, ausgeführt werden. Wenn die **A**-, ..., **E**-Tasten durch ein Programm anders definiert wurden, sind diese Funktionen immer noch über zwei andere Tasten verfügbar.

Beispiel: Schalten Sie den Rechner aus, dann ein, speichern Sie 4, 3, 2 und 1 in die Stack-Register **T**, **Z**, **Y** und **X**. Sehen Sie sich jetzt den Inhalt des Stacks unter Verwendung von **xzy**, **E**, **R↓**, **D** und **R↑** an.

T						4	4
Z				4	4	3	3
Y		4	4	3	3	2	2
X	4	4	3	3	2	2	1
Taste	[4]	↑	[3]	↑	[2]	↑	[1]

ENTER↑ ist hier abgekürzt als **↑**.

T	4	4	4	4
Z	3	3	3	3
Y	1	2	1	2
X	2	1	2	1

Taste

xzy

xzy

E E

Verwendung von xzy

Verwendung von E

Beachten Sie, daß die zweimalige Verwendung von xzy oder E den Stack in seiner ursprünglichen Anordnung beläßt.

T	1	2	3	4	1	2	3	4
Z	4	1	2	3	4	1	2	3
Y	3	4	1	2	3	4	1	2
X	2	3	4	1	2	3	4	1

Taste

R↑

R↑

R↑

R↑

D D D D

Verwendung von R↑

Verwendung von D

Beachten Sie, daß die viermalige Verwendung von R↑ oder D den Stack in seiner ursprünglichen Anordnung beläßt.

T	3	2	1	4
Z	2	1	4	3
Y	1	4	3	2
X	4	3	2	1

Taste

R↑

R↑

R↑

R↑

Beachten Sie, daß die viermalige Verwendung von R↑ den Stack in seiner ursprünglichen Anordnung beläßt.

AUFRUFEN DER ZAHL π

π ist in Ihrem HP-65 als Konstante fest vorprogrammiert. Wenn Sie diesen Wert in einer Rechnung benötigen, drücken Sie einfach **9** **π** .

Beispiel. Berechnen Sie die Fläche eines Kreises mit dem Radius 3. Kreisfläche = $\pi 3^2$.

Drücken Sie

9 **π**

3 **ENTER** **x**

x

Anzeige

3.14

9.00

28.27

Bemerkung

Ruft π nach X

Berechnet 3×3

Ergebnis $9 \times \pi$

EINE WEITERE VERWENDUNGSMÖGLICHKEIT IHRES HP-65 – ZINSESZINSTABELLE

Die HP-Taschenrechner bieten Ihnen viele Möglichkeiten zur Erschließung neuer Anwendungsgebiete und Lösungsmethoden. Oft haben uns unsere Kunden mit besonders eleganten Lösungsverfahren überrascht, bei denen sie sich die speziellen Eigenschaften ihrer Rechner zu Nutzen machten. Als Beispiel stellen wir Ihnen hier die Berechnung einer geometrischen Folge vor, die den verzinsten Betrag einer einmaligen Investition nach Ablauf der jeweiligen Perioden angibt.

EINE ZINSESZINSTABELLE

Unter einer geometrischen Folge versteht man eine solche Folge von Zahlen, bei der sich jedes Glied aus dem vorhergehenden durch Multiplikation mit einem konstanten Faktor ergibt. Beispiel: 4, 8, 16, 32 usw.; der konstante Faktor ist hier 2. Auch das Wachstum einer Investition von DM 1000,- zu 10% pro Periode stellt eine geometrische Folge dar. Das erste Glied ist in diesem Falle 1000 und der Wachstumsfaktor 1.10. Die Berechnung der einzelnen Glieder dieser Folge führte unser Kunde auf folgende Weise durch:

Drücken Sie	Anzeige	Bemerkung
1.10 ENTER ENTER		
ENTER 1000	1000.00	Anfangsbetrag
x	1100.00	Betrag nach 1. Periode
x	1210.00	Betrag nach 2 Perioden
x	1331.00	Betrag nach 3 Perioden
x	1464.10	Betrag nach 4 Perioden
x	1610.51	Betrag nach 5 Perioden
x	1771.56	Betrag nach 6 Perioden

Mit anderen Worten, geben Sie den Wachstumsfaktor (1.10) in die Stack-Register **Y**, **Z** und **T** und tasten Sie den Anfangsbetrag (1000) in das **X**-Register. Das jeweils nächste Glied der Folge berechnen Sie jetzt durch einfaches Drücken von **x**. Wenn Sie beispielsweise die **x**-Taste zum ersten Mal drücken, berechnen Sie 1000×1.10 ($x \times y$). Das Ergebnis (1100.00) wird im **X**-Register angezeigt und durch die Verschiebung des Stack nach unten eine neue Kopie des Wachstumsfaktors (1.10) erzeugt. Da eine Kopie dieses Faktors immer dann entsteht, wenn der Stack nach unten verschoben wird, brauchen Sie ihn nie erneut einzugeben.

Inhalt	Register	
t (1.10)	T (1.10)	Inhalte der Stack-Register vor und nach der ersten Multiplikation
z (1.10)	Z (1.10)	
y (1.10)	Y (1.10)	
x (1000)	X (1100)	
	($x \times y$)	

FUNKTIONEN

Sie haben bereits die Anwendung der arithmetischen Funktionen (+, -, ×, ÷) im Stack und in den adressierbaren Registern kennengelernt. Ebenso haben Sie erfahren, wie die Inhalte der verschiedenen Register des Rechners verschoben und umgespeichert werden können und wie Daten in Festkomma- und Exponentialdarstellung (wissenschaftliche Schreibweise) eingegeben und angezeigt werden. Um das Thema der manuellen Durchführung von Rechnungen abzuschließen, wollen wir jetzt auf die nicht-arithmetischen Funktionen zurückkommen, also auf Funktionen wie Sinus, Logarithmus, Quadratwurzel usw.

Folgende Tasten werden in diesem Abschnitt vorgestellt:

ABS	→DMS	INT	n!	R→P	1/x
COS	D.MS+	LN	→OCT	SIN	√x
DEG	GRD	LOG	RAD	TAN	y ^x

Diese Funktionen sind wichtig und wirkungsvoll. Auch wenn Sie an der direkten Verwendung der verfügbaren Funktionen wenig interessiert sind, so nutzen Sie sie doch indirekt bei der Verwendung vorprogrammierter Magnetkarten aus dem Standard-Paket oder anderen Anwendungs-Paketen. Falls Sie das persönliche Investitions-Programm im Rahmen der Einleitung dieses Handbuches probeweise gerechnet haben, haben Sie beispielsweise bereits die Funktionen y^x (Exponentialfunktion) und INT (ganzzahliger Teil von x) genutzt. Ohne Verwendung von y^x hätte das Programm auf die wiederholte Anwendung wesentlich aufwendigerer Rechenschritte zurückgreifen müssen.

Die Funktionen sind äußerst einfach anzuwenden. Sie haben bereits in der Einleitung erfahren, daß zur Ausführung einer der Funktionen zuerst eine Präfix-Taste (**f**, **f⁻¹** oder **■**) und anschließend die gewünschte Funktions-Taste zu drücken ist.

Die Präfix-Taste **■** ist zu drücken, wenn eine Funktion mit blauem Symbol aufgerufen werden soll, die Taste **f** zur Ausführung einer goldfarbenen Funktion. Wird als Präfix die Taste **f⁻¹** gedrückt, so wird die inverse oder komplementäre Funktion zu dem goldfarbenen Symbol ausgeführt.

Vielleicht haben Sie unter Berücksichtigung dieser Regel bereits einige dieser Funktionen angewendet. Sie sollen jetzt einen systematischen Überblick über die verschiedenen verfügbaren Funktionen erhalten und die bei ihrer Anwendung zu berücksichtigenden Einzelheiten erfahren. Das Wesentliche über diese Funktionen ist in den Tabellen 3-1, 3-2 und 3-3 zusammengefaßt.

Zu einer gegebenen Funktion sind in der entsprechenden Tabelle alle Bedingungen aufgeführt, denen die Ausgangswerte entsprechen müssen, die zu benutzenden Tasten angegeben und Einzelheiten über das bzw. die Ergebnisse genannt. Wenn Sie mit dem Rechnen sofort beginnen wollen, können Sie diesen Abschnitt mit den Tabellen abschließen und die Beispiele auslassen.

FUNKTIONEN, DIE SICH AUF WINKEL BEZIEHEN

Diese Funktionen sind in Tabelle 3-1 zusammengestellt. Sie umfassen die trigonometrischen Funktionen (Sinus, Kosinus, Tangens und die Umkehrfunktionen), die Umwandlung rechtwinkliger in Polar-Koordinaten und umgekehrt, die Addition und Subtraktion von Winkeln, die in Grad, Minuten und Sekunden ausgedrückt sind und die Umwandlung dezimal gegebener Winkel in/von Grad, Minuten und Sekunden.




Tasten	Funktion	Ausgangswert(e)	Ergebnis(se)
f COS	Kosinus	Winkel ¹	Kosinus (x) in X
f-1 COS	Arkuskosinus	x muß kleiner (oder gleich) 1 oder größer (oder gleich) -1 sein ($ x \leq 1$)	Hauptwert von $\arccos(x)$ in X ($0^\circ \leq \text{Resultat} \leq 180^\circ$) ²
f SIN	Sinus	Winkel ¹	Sinus (x) in X
f-1 SIN	Arkussinus	x muß kleiner (oder gleich) 1 oder größer (oder gleich) -1 sein ($ x \leq 1$)	Hauptwert von $\arcsin(x)$ in X ($-90^\circ \leq \text{Resultat} \leq 90^\circ$) ²
f TAN	Tangens	Winkel ¹	Tangens (x) in X
f-1 TAN	Arkustangens	x nicht beschränkt	Hauptwert von $\arctan(x)$ in X ($-90^\circ \leq \text{Resultat} \leq 90^\circ$) ²
f R→P	Umwandlung rechtwinkliger Koordinaten (x, y) in Polarkoordinaten (r, θ)	x, y in X, Y	r, θ in X, Y
f-1 R→P	Umwandlung von Polarkoordinaten (r, θ) in rechtwinkliger Koordinaten (x, y)	r, θ ¹ in X, Y	x, y in X, Y
f →DMS	Wandelt dezimale Winkel in die Form DDDDD.MMSS um ³	dezimaler Winkel ⁴	DDDDD.MMSS in X
f-1 →DMS	Wandelt DDDDD.MMSS Winkel in dezimale Form um ³	DDDDD.MMSS	dezimaler Winkel ⁴ in X
f DMS+	Addiert (x+y) in der Form DDDDD.MMSS ³	$\left. \begin{matrix} y \\ x \end{matrix} \right\} \text{DDDDD.MMSS}^4$	DDDDD.MMSS in X (Summe) ⁴
f-1 DMS+	Subtrahiert (y-x) in der Form DDDDD.MMSS ³	$\left. \begin{matrix} y \\ x \end{matrix} \right\} \text{DDDDD.MMSS}^4$	DDDDD.MMSS in X (Differenz) ⁴

¹ Dezimaler Winkel im eingestellten Winkel-Modus² Oder entsprechend in Neugrad oder rad³ DDDDD.MMSS Format. D = Grad, MM = Minuten, SS = Sekunden⁴ Der Winkel darf nicht größer sein als 99999.99999 dezimale Grad (oder entsprechend in rad oder Neugrad) oder 99999.59599 im DDDDD.MMSS**Tabelle 3-1.** Funktionen, die sich auf Winkel beziehen.

WINKEL-MODUS

Alle Funktionen, die sich auf Winkel beziehen, setzen voraus, daß diese in der Einheit des vorherrschenden Winkel-Modus ausgedrückt werden. Beim Einschalten des Rechners wird der Winkel-Modus automatisch auf «Grad dezimal» gestellt. Den Winkel-Modus können Sie auf Wunsch in «rad» (Bogenmaß), Neugrad (Gon) oder «Grad dezimal» einstellen, indem Sie die Funktionen für die Wahl des Winkel-Modus verwenden.

Wahl des Winkel-Modus

Tasten	Funktion
 GRD	Stellt Modus auf Neugrad
 RAD	Stellt Modus auf rad (Bogenmaß)
 DEG	Stellt Modus auf Grad (Altgrad)


$$400 \text{ Gon (Neugrad)} = 360 \text{ (Alt-) Grad} = 2\pi \text{ rad}$$

Der Winkel-Modus bezieht sich auf die folgenden Tasten:

SIN, **COS**, **TAN**, **R↔P**, **↔DMS**

In den Beispielen wird der Winkel-Modus als Grad angenommen, wenn nichts anderes angegeben ist.

GRAD, MINUTEN, SEKUNDEN

Die dezimale Darstellung eines Winkels können Sie in die Form Grad, Minuten, Sekunden umwandeln und umgekehrt. Das Format für Grad, Minuten, Sekunden ist DDDDD.MMSS. Stellen Sie also zur Anzeige dieses Formates den Rechner mit **DSP**  **4** auf vier Nachkommastellen ein. Im nachfolgenden Beispiel ist gezeigt, wie diese Funktion vom Winkel-Modus abhängt.

Beispiel, Teil 1. Wandeln Sie $\pi/7$ rad in Grad, Minuten, Sekunden um.

Drücken Sie

DSP  **4**

Anzeige

0.4488

Bemerkung

Stellt Anzeige auf vier Nachkommastellen ein

 **π** **7** 

$\pi/7$

42 Funktionen

g RAD	0.4488	Stellt Winkel-Modus auf rad
f →DMS	25.4251	Ergebnis: 25° 42' 51"

Beispiel, Teil 2. Verfahren Sie jetzt umgekehrt, aber wandeln Sie den Winkel in Neugrad (anstatt rad) um. Beachten Sie: Dieses Verfahren kann zur Umwandlung zwischen den verschiedenen Winkleinheiten dienen, d.h.

Grad \longleftrightarrow rad (Bogenmaß)

Grad \longleftrightarrow Neugrad

rad \longleftrightarrow Neugrad

Drücken Sie	Anzeige	Bemerkung
g GRD	25.4251	Stellt Winkel-Modus auf Neugrad
f1 →DMS	28.5713	Ergebnis in Neugrad

Beispiel, Teil 3. Wandeln Sie jetzt in Grad dezimal um.

Drücken Sie	Anzeige	Bemerkung
g LST x	25.4251	25° 42' 51"
g DEG	25.4251	Stellt Winkel-Modus auf Grad
f1 →DMS	25.7142	Ergebnis in Grad dezimal
DSP □ 2	25.71	Stellt Anzeige zurück

Beispiel: Addition/Subtraktion von DDDDD.MMSS. Ermitteln Sie die Summe von 45° 10' 50" und 44° 49' 10".

Drücken Sie	Anzeige	Bemerkung
DSP □ 4	0.0000	
45.1050	45.1050	
ENTER +	45.1050	
44.4910	44.4910	
f D.MS+	90.0000	Ergebnis: 90° 00' 00"

Ein Musikstück beginnt um 9:25' 7" und endet um 9:39' 47". Wie lange dauert es?

Drücken Sie	Anzeige	Bemerkung
DSP □ 4	0.0000	
9.3947	9.3947	Uhrzeit am Ende

ENTER ↑
 9.2507
f-1 **DMS+**
DSP **□** **2**

9.3947
 9.2507
 0.1440
 0.14

Uhrzeit bei Beginn
 Ergebnis: 14' 40" Dauer
 Stellt Anzeige auf zwei
 Dezimalstellen zurück

Beispiel: Trigonometrische Funktionen. Berechnen Sie den Kosinus von 60° .

Drücken Sie

□ **DEG** 60
f **COS**

Anzeige

60.

0.50

Bemerkung

Ergebnis

Berechnen Sie den Arkuskosinus von $(-1.)$ in rad.

Drücken Sie

□ **RAD** 1 **CHS**
f-1 **COS**

Anzeige

-1.

3.14

Bemerkung

Ergebnis in rad

Berechnen Sie den Sinus von 30° .

Drücken Sie

□ **DEG** 30
f **SIN**

Anzeige

30.

0.50

Bemerkung

Ergebnis

Berechnen Sie den Arkussinus von (1) in rad.

Drücken Sie

□ **RAD** 1
f-1 **SIN**

Anzeige

1.

1.57

Bemerkung

Ergebnis in rad

Berechnen Sie den Tangens von 45° .

Drücken Sie

□ **DEG** 45
f **TAN**

Anzeige

45.

1.00

Bemerkung

Ergebnis

Berechnen Sie den Arkustangens von (39.4) in rad.

Drücken Sie

□ **RAD** 39.4
f-1 **TAN**

Anzeige

39.4

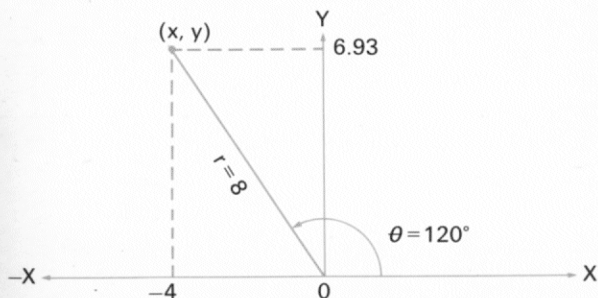
1.55

Bemerkung

Ergebnis in rad

44 Funktionen

Beispiel: Polarkoordinaten in rechtwinklige Koordinaten. Wandeln Sie Polarkoordinaten ($r=8$, $\theta=120^\circ$) in rechtwinklige (kartesische) Koordinaten um:



Drücken Sie

$\boxed{\text{DEG}}$

120 $\boxed{\text{ENTER}}$

8

$\boxed{\text{R}\rightarrow\text{P}}$

$\boxed{\text{x}\rightarrow\text{y}}$

Anzeige

0.00

120.00

8.

-4.00

6.93

Bemerkung

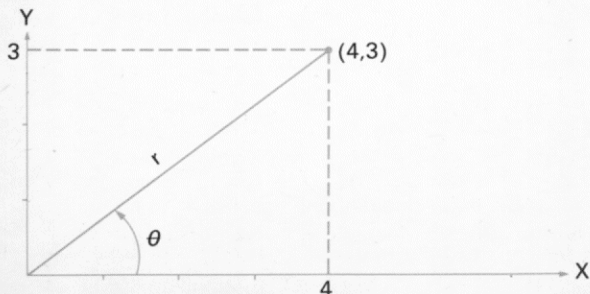
θ

r

x-Koordinate

y-Koordinate

Beispiel: Rechtwinklige Koordinaten in Polarkoordinaten. Wandeln Sie rechtwinklige (kartesische) Koordinaten ($x=4$, $y=3$) in Polarkoordinaten um; drücken Sie den Winkel in Grad aus:



Drücken Sie	Anzeige	Bemerkung
g DEG 3 ENTER ↑	3.00	y-Koordinate
4	4.	x-Koordinate
f R→P	5.00	r (Radius)
g x↔y	36.81	θ (Winkel in Grad)

UMWANDLUNGEN

Die Umwandlungen sind in Tabelle 3-2 aufgeführt. Sowohl die Ausgangswerte als auch die Resultate der Umwandlungen stehen im **X**-Register. Beachten Sie, daß die Winkel-Umwandlungen in Tabelle 3-1 stehen.

Beispiel: Umwandlung von oktal in dezimal. Viele Computer rechnen mit Zahlen in Oktal-Darstellung (zur Basis 8), anstatt mit solchen in Dezimal-Darstellung (zur Basis 10). Die Funktion **→OCT** Ihres HP-65 erlaubt Ihnen die einfache Umwandlung von Zahlen zwischen beiden Darstellungen. Ermitteln Sie beispielsweise die zu der dezimalen Zahl 512 (512_{10}) äquivalente oktale Zahl.

Drücken Sie	Anzeige	Bemerkung
512 f →OCT	1000.00	Oktaldarstellung von 512_{10}

Wandeln Sie die oktale Zahl 2000 in ihr dezimales Äquivalent um:

2000 f1 →OCT	1024.00	Dezimaldarstellung von 2000_8
----------------------------	---------	---------------------------------

Tasten	Funktion	Ausgangswert(e)	Ergebnis(se)
f →OCT	Wandelt dezimale ganze Zahl in Oktal-Darstellung um (Basis 8)	x_{10} eine dezimale ganze Zahl, kleiner als 1073741824_{10}	x_8 in X
f⁻¹ →OCT	Wandelt oktale ganze Zahl in dezimale Form (Basis 10) um	x_8 eine oktale ganze Zahl	x_{10} in X
f INT	Beschneidet x zu einer ganzen Zahl mit Vorzeichen	\pm ganze Zahl. Dezimalteil in X	\pm ganze Zahl .0 in X
f⁻¹ INT	Beschneidet x zu dem Dezimalteil von x , beläßt das Vorzeichen	\pm ganze Zahl. Dezimalteil in X	\pm 0. Dezimalteil in X
■ ABS	Absolutwert von x	$\pm x$	Falls x negativ, $-x$ in X ; sonst keine Änderung

Tabelle 3-2. Umwandlungen von **x**.

Beispiel: Abschneiden einer Zahl vor/hinter dem Dezimalpunkt. Die Daten für das persönliche Investitions-Programm (siehe Einleitung) waren in dem Format mm.yyyy einzugeben. Das Programm trennt mm von yyyy mit Hilfe der **[INT]**-Funktionen. Führen Sie das gleiche jetzt für das Datum 12.1980 durch.

Drücken Sie	Anzeige	Bemerkung
[DSP] [.] [4]	0.0000	
12.1980	12.1980	
[f] [INT]	12.0000	Ergebnis: ganzzahliger Teil
[g] [LST X]	12.1980	Rückruf des ursprünglichen Wertes
[f1] [INT]	0.1980	Ergebnis: Dezimalteil der Zahl
[DSP] [.] [2]	0.20	Stellt Anzeige zurück

Beispiel: Absolutwert. Manche Rechnungen benötigen die Absolutbeträge von Zahlen. Vom Tastenfeld aus genügt es, die Zahl zu beobachten und bei einer negativen Zahl das Vorzeichen zu wechseln (unter Verwendung von **[CHS]**). Im Rahmen eines Programms verwenden Sie dazu die **[ABS]**-Funktion, die automatisch das Vorzeichen einer negativen Zahl ändert. Berechnen Sie als Beispiel die Absolutwerte von 3 und -3.

Drücken Sie	Anzeige	Bemerkung
3 [g] [ABS]	3.00	+3
[CHS]	-3.00	
[g] [ABS]	3.00	-3

FUNKTIONEN VON x UND DIE ALLGEMEINE EXPONENTIALFUNKTION (y^x)

Diese Funktionen sind in Tabelle 3-3 zusammengefaßt. Das Resultat dieser Funktionen steht jeweils im **X**-Register. Mit Ausnahme der Exponentialfunktion beziehen sich die Funktionen auch ausschließlich auf **x**; **[y^x]** erwartet zusätzlich ein **y** im **Y**-Register. Es ist erwähnenswert, daß sich die in der Spalte «Ausgangswert(e)» angeführten Bedingungen bei

Tasten	Funktion	Ausgangswert(e)	Ergebnis(se)
f LN	Natürlicher Logarithmus (Basis e)	x darf nicht gleich oder kleiner als Null sein ($x > 0$)	$\ln(x)$ in X
f⁻¹ LN	Exponentialfunktion (Basis e) (e^x)	x nicht beschränkt	e^x in X
f LOG	Dekadischer Logarithmus (Basis 10)	x darf nicht gleich oder kleiner als Null sein ($x > 0$)	$\log(x)$ in X
f⁻¹ LOG	Exponentialfunktion (Basis 10) (10^x)	x nicht beschränkt	10^x in X
f \sqrt{x}	Quadratwurzel (\sqrt{x})	x nicht negativ ($x \geq 0$)	\sqrt{x} in X
f⁻¹ \sqrt{x}	Quadrat (x^2)	x nicht beschränkt	x^2 in X
$\frac{1}{x}$	Reziprokwert ($1/x$)	x nicht gleich 0 ($x \neq 0$)	$1/x$ in X
$n!$	Fakultät ($n!$) $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (n-1) \cdot n$, $0! = 1$	positives ganzzahliges n in X	$n!$ in X
y^x	Allgemeine Exponential- funktion (y^x)	($x \geq 0$; x ganzzahlig) x nicht beschränkt, y positiv ($y > 0$)	y^x in X ; Stack wird nach unten geschoben

Tabelle 3-3. Funktionen von x und die allgemeine Exponentialfunktion.

einigem Nachdenken von selbst ergeben. So heißt es beispielsweise in der Tabelle, daß bei der Berechnung des Reziprokwertes x ungleich Null sein muß. Dies bedeutet nichts weiter, als daß eine Division durch Null ($x \div 0$) unsinnig und verboten ist. Wenn wir uns dennoch nicht daran halten und eine solche Operation durchführen, warnt uns die blinkende Anzeige sofort vor dem Fehler. Probieren Sie es doch spaßeshalber einmal aus; drücken Sie **CLX** **0** **$\frac{1}{x}$** . Das Blinken der Anzeige können Sie durch Drücken einer beliebigen Taste stoppen.

Beispiel: Dekadischer Logarithmus. Berechnen Sie die Leistungsverstärkung eines Verstärkers in Dezibel, der die Eingangsleistung verdoppelt.

Anmerkung: dezibel (dB) = $10 \log (2)$

Drücken Sie	Anzeige	Bemerkung
10 ENTER	10.00	Speichern des Wertes 10
2 f LOG	.30	$\log (2)$
x	3.01	Resultat

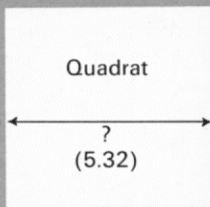
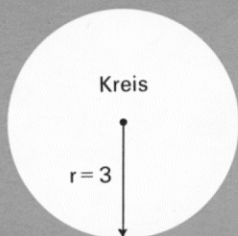
Beispiel: Exponentialfunktion e^x . Zeigen Sie die Konstante e mit neun Stellen an ($e = e^1$).

Drücken Sie	Anzeige	Bemerkung
1 f1 LN	2.72	
DSP • 9	2.718281828	Ergebnis
DSP • 2	2.72	Stellt Anzeige zurück

Beispiel: Quadrat und Quadratwurzel. Welche Seitenlänge hat ein Quadrat, dessen Fläche gleich der eines Kreises mit Radius $r = 3$ ist?

$\pi \times 3^2$ ist die Fläche des Kreises. Die Quadratwurzel aus diesem Wert ergibt die Seitenlänge des flächengleichen Quadrates.

Quadratur des Kreises



Drücken Sie



Anzeige

3.14
9.00
28.27
5.32

Bemerkung

π
 3^2
Kreisfläche
Seitenlänge des Quadrates

Beispiel: Reziprokwert. Berechnen Sie: $1/4 = .25$.

Drücken Sie



Anzeige

0.25

Bemerkung

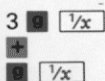
Reziprokwert von 4

Sie können diesen Wert natürlich in einer weiteren Rechnung verwenden. Fahren Sie beispielsweise fort und berechnen Sie

$$\frac{1}{\frac{1}{4} + \frac{1}{3}}$$

$\frac{1}{4}$ ist bereits berechnet, also

Drücken Sie



Anzeige

0.25
0.33
0.58
1.71

Bemerkung

Reziprokwert von 4
Reziprokwert von 3
Summe der Reziprokwerte
Ergebnis: Reziprokwert der Summe

Beispiel: Fakultät. Berechnen Sie die Anzahl der Möglichkeiten, wie sich sechs Personen für eine Gruppenaufnahme in eine Reihe aufstellen können.

Drücken Sie6 **9** **n!****Anzeige**

720.00

Bemerkung

Ergebnis

Beispiel: Exponentialfunktion y^x . In dem vorangehenden Abschnitt haben wir die Folgeglieder einer geometrischen Folge berechnet und herausgefunden, daß ein Kapital von DM 1000,— bei 10% Verzinsung nach 6 Perioden auf DM 1771,56 angewachsen ist. Unter Verwendung der **y^x**-Funktion kann das gleiche Ergebnis wie folgt erhalten werden:

1000 (1.10)⁶**Drücken Sie**1000 **ENTER** **↑**1.10 **ENTER** **↑** 6**9** **y^x****x****Anzeige**

1000.00

Bemerkung

Anfangskapital

1.77

(1.10)⁶

1771.56

Ergebnis

PROGRAMMIERUNG

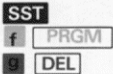
Sicherlich war beim Kauf Ihres HP-65 auch für Sie dessen herausragende Eigenschaft ausschlaggebend – freie Programmierbarkeit. Sie haben jetzt den Abschnitt dieses Handbuches erreicht, der in die vielfältigen Möglichkeiten der Programmierung des HP-65 einführt. Dabei werden Sie bald erkennen, daß die Tastenfeld-Programmiersprache des HP-65 weder kompliziert noch schwierig zu erlernen ist. Sie sollten sich die Zeit nehmen und alle in diesem Abschnitt aufgeführten Programmbeispiele durcharbeiten. Auf diese Weise werden Sie von einfachen Programmen ausgehend – ein solches haben Sie bereits in der Einleitung erstellt – im Laufe dieses Abschnitts die Fertigkeit erlangen, so anspruchsvolle Programme zu schreiben, wie Sie sie in den Anwendungs-Paketen finden.

WAS IST EIN PROGRAMM?

Ein Programm ist nichts weiter als eine im Rechner gespeicherte Folge von Tastenbefehlen, die auf den Druck einer einzigen Taste automatisch vom Rechner ausgeführt wird – ein Tastendruck ersetzt eine ganze Tastenfolge! In den vorangegangenen Abschnitten dieses Handbuches wurden, sooft Beispiele zu rechnen waren, Sie, die Bedienungsperson, programmiert. Um ein bestimmtes Ergebnis zu erhalten, wurden Sie aufgefordert, in einer vorgegebenen Reihenfolge Tasten zu drücken. In aller Regel war auch das Resultat falsch, wenn Sie sich nicht an die genau vorgegebene Tastenfolge gehalten haben. In gleicher Weise wird auch im Rahmen eines Programms dem Rechner eine bestimmte Tastenfolge eingegeben. Der Rechner speichert sie und kann sie anschließend beliebig oft ausführen, wobei er dann «unter eigener Regie» wesentlich schneller arbeitet, als Sie das von Hand könnten.

Welche Tastenfolge geben Sie nun in den Rechner ein? Der Kern eines jeden Programms, das Sie schreiben, besteht aus den Tastenbefehlen, mit denen Sie auch im RUN-Modus das Problem manuell lösen würden.

Tatsächlich gibt es unter allen Tastenbefehlen nur drei Kombinationen, die nicht zur späteren Ausführung im Rechner gespeichert werden können:



Diese drei Tastenfolgen sind die einzigen aktiven Operationen im W/PRGM-Modus. Jeder andere Tastendruck im W/PRGM-Modus wird zur späteren Ausführung gespeichert.

Wie Sie wissen führt im RUN-Modus das Drücken einer jeden Taste zur Ausführung einer ganz bestimmten Operation. Die Ausführung dieser Operation kann allerdings auf zwei verschiedene Weisen veranlaßt werden: Manuell vom Tastenfeld oder vom Programmspeicher aus (sofern die Tastenfolge dort zuvor abgespeichert wurde). Eine Ausnahme bilden die folgenden Tasten, die in Abhängigkeit davon, ob sie vom Tastenfeld aus gedrückt oder vom Programm ausgeführt werden, unterschiedliche Wirkung haben:



Diese Befehle steuern den Programmablauf. Ihre unterschiedliche Wirkung sollten Sie genau beachten.

PROGRAMMIERBARE FUNKTIONEN

Sie haben bereits an anderer Stelle erfahren, daß fünf Operationen auf zwei verschiedene Weisen ausgeführt werden können. Sie können die Tasten **g** $\frac{1}{x}$ drücken oder einfach **A**; **f** \sqrt{x} oder einfach **B**; und so weiter. Die fünf Tasten **A**

54 Programmierung

bis **E** werden zur Steuerung der Programmausführung verwendet, wobei jede Taste über das Programm definiert ist, das durch sie aufgerufen wird. Wenn Sie den Rechner einschalten, werden automatisch fünf «Ersatzprogramme» in den Speicher geladen und diesen Tasten zugeordnet. Die Funktion dieser Programme ist jeweils über der entsprechenden Taste als weißes Symbol angegeben. Die Verwendung dieser Tasten beim manuellen Rechnen ist sehr bequem, da Sie die entsprechenden Funktionen jetzt mit nur einem Tastendruck ausführen können (z. B. **A** anstatt **g** \sqrt{x}). Die Bedeutung der Tasten **A** bis **E** können Sie durch andere Programme beliebig abändern. Das kurze Programm, das Sie im Rahmen der Einleitung erstellt haben, zeigt Ihnen am Beispiel, wie dies geschieht. In diesem Fall wurde die Taste **A** dazu definiert, die dritte Potenz einer Zahl zu berechnen.

DER PROGRAMMSPEICHER

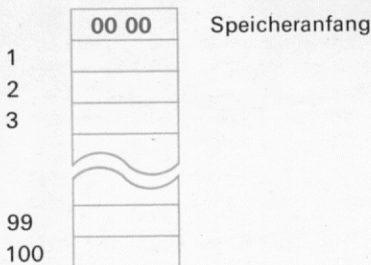
Sie sollen jetzt, von diesen «Ersatz-Programmen» ausgehend, etwas mehr über den Programmspeicher erfahren. Schalten Sie Ihren HP-65 zuerst aus, dann ein. Die Tasten **A** bis **E** sind jetzt mit den «Ersatz-Programmen» belegt. Schieben Sie jetzt den Modusschalter in die Stellung W/PRGM (Write PRoGrAm). In der Anzeige sehen Sie:



Markierung für Speicheranfang

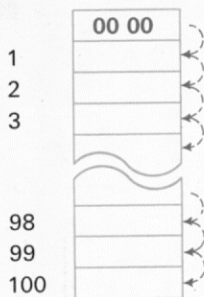
Diese Anzeige ist das Zeichen dafür, daß Sie sich am Anfang des Programmspeichers befinden. Er verfügt über 100 nutzbare Speicherzeilen und die Markierung für den Speicheranfang. In der nachfolgenden Zeichnung ist der Aufbau des Programmspeichers graphisch dargestellt. Beachten Sie dabei, daß die Markierung des Speicheranfangs zwar eine Speicherzeile belegt (nicht eine der 100 frei verfügbaren), in dieser Zeile aber keine Tastenbefehle gespeichert werden können.

Die übrigen Speicherzeilen können jeweils einen oder in bestimmten Fällen auch zwei Tastenbefehle aufnehmen.



DER PROGRAMMSCHRITT-ANZEIGER*

Während des Programmablaufs geht der Rechner, vom «Programmschritt-Anzeiger» gesteuert, den Speicher von oben nach unten durch und führt Schritt um Schritt aus.



Dabei gibt der Programmschritt-Anzeiger jeweils den nächsten auszuführenden Programmschritt an.

EINZELSCHRITT

SST (Single STep, Einzelschritt) kann nicht im Programmspeicher gespeichert werden, sondern dient im W/PRGM-

* Für eine fortgeschrittene Programmierung ist es von großer Bedeutung, daß Sie die Details im Zusammenhang mit der Wirkungsweise des Programmschritt-Anzeigers verstehen. Sie finden nähere Einzelheiten darüber im Anhang B.

56 Programmierung

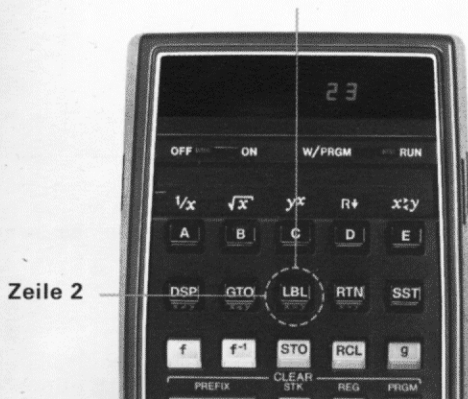
Modus zur schrittweisen Anzeige der einzelnen Programmschritte. Der Programmschritt-Anzeiger wird mit jedem Drücken von **SST** einen Schritt weitergerückt, wobei der nächste Programmschritt angezeigt, aber nicht ausgeführt wird. Die ebenfalls mögliche Verwendung von **SST** im RUN-Modus wird an späterer Stelle behandelt.

TASTEN-CODE

Wir wollen uns jetzt mit Hilfe der Taste **SST** das Programm ansehen, das der Taste **A** zugeordnet ist. Drücken Sie **SST** einmal, als Anzeige erhalten Sie: 23.

Dies ist der Code für den ersten Programmschritt. Die erste Zahl gibt die Zeile und die zweite Zahl die Spalte als Angabe für die Position auf dem Tastenfeld an. Im Beispiel besagt der Code 23, daß es sich um die Taste in der zweiten Zeile und dritten Spalte auf dem Tastenfeld, also um die Taste **LBL** (LaBeL, Marke) handelt.

Spalte 3



Die einzige Ausnahme bei der Festlegung dieses Codes bilden die Zifferntasten **0** bis **9**, die der besseren Erkennbarkeit

wegen als 00 bis 09 angezeigt werden. Das gleiche gilt für die Funktionen, die diese Tasten nach vorangegangenen Präfix **g**, **f** oder **f1** haben (entsprechend den blauen oder goldfarbenen Symbolen).

Drücken Sie erneut **SST**, die Anzeige ändert sich in: 11.

Dieser Code bezeichnet die **A**-Taste (erste Zeile, erste Spalte). Nach nochmaligem Drücken von **SST** wird der Code der blauen Präfixtaste **g** angezeigt: 35.

Auf das nächste Drücken von **SST** ändert sich die Anzeige in: 04.

Beachten Sie an dieser Stelle, daß, da der vorangegangene Code für die blaue Präfixtaste **g** stand, der Rechner den Code 04 als $\frac{1}{x}$ interpretiert, also als die blaue Alternativfunktion der Taste **4**. Wenn Sie schließlich **SST** ein weiteres Mal drücken, sehen Sie in der Anzeige den letzten Tastencode des Programms, das zu der **A**-Taste gehört: **RTN** (ReTurN, Zurück!) 24.

Also sieht das vollständige Programm, das auf Drücken der Taste **A** ausgeführt wird, wie folgt aus:

Tastencode	Taste
23	LBL
11	A
35	g
04	$\frac{1}{x}$
24	RTN

Bemerkung

Hier beginnt die Ausführung des Programms, wenn **A** gedrückt wurde.

Diese Tasten haben im Programm die gleiche Wirkung wie vom Tastenfeld aus.

Bezeichnet das Ende des Programms.

Fahren Sie jetzt mit dem Drücken von **SST** fort und sehen Sie sich die Programme für die übrigen Tasten **B** und **C** an. Die Tastenfolge und die entsprechenden Codes sind nachstehend angegeben:

Tastencode	Taste
23	LBL
12	B
31	f
09	\sqrt{x}
24	RTN
23	LBL
13	C
35	g
05	y^x
24	RTN

Bemerkung

Hier beginnt die Ausführung nach Drücken von **B**.

Auch hier wieder haben diese Schritte die gleiche Funktion wie vom Tastenfeld aus.

Bezeichnet das Ende des Programms.

Hier beginnt die Ausführung nach Drücken von **C**.

Berechnet y^x , wie das auch vom Tastenfeld aus der Fall wäre.

Bezeichnet das Ende des Programms.

KOMBINIERTER CODES

Um wertvollen Speicherplatz einzusparen, sind die Codes der am häufigsten verwendeten Tastenpaare, bestehend aus einer Präfixtaste und einer Folgetaste, zu einem einzigen kombinierten Code zusammengefaßt (aufgrund interner Beschränkungen ist dies nicht für alle solchen Kombinationen möglich). Ein Beispiel dafür finden Sie in dem zur Taste **D** gehörenden «Ersatz-Programm». Falls Sie noch nicht bei **D** sind, tasten Sie sich schrittweise mit **SST** durch den Speicher, bis Sie diese Stelle erreichen. Das Programm sieht wie folgt aus:

Tastencode	Taste
23	LBL
14	D
35 08	g R↓
24	RTN

Bemerkung

Hier beginnt die Ausführung nach Drücken von **D**.

Wirkung wie vom Tastenfeld aus.

Bezeichnet das Ende des Programms.

Sie sehen, wie die Tasten **g** und **R↓** zusammengefaßt und als kombinierter Code 35 08 gespeichert werden.

Fahren Sie mit dem Drücken der Taste **SST** fort und sehen Sie sich auch das Programm unter **E** an, das ebenfalls einen

kombinierten Code beinhaltet. Die Tastenfolge und die entsprechenden Codes sind nachstehend angegeben:

Tastencode	Taste
23	LBL
15	E
35 07	g x\leftrightarrowy
24	RTN

Bemerkung

Hier beginnt die Ausführung nach Drücken von **E**.

Wirkung wie vom Tastenfeld aus.

Bezeichnet das Ende des Programms.

Nachstehend sind die Tastenpaare mit kombiniertem Code aufgeführt:

Tastencode	Tasten	Tastencode	Tasten
35 00	g LST x	35 09	g R\uparrow
35 07	g x\leftrightarrowy	35 08	g R\downarrow
33 01	STO 1	34 01	RCL 1
33 02	STO 2	34 02	RCL 2
33 03	STO 3	34 03	RCL 3
33 04	STO 4	34 04	RCL 4
33 05	STO 5	34 05	RCL 5
33 06	STO 6	34 06	RCL 6
33 07	STO 7	34 07	RCL 7
33 08	STO 8	34 08	RCL 8
35 01	g NOP		

Beachten Sie insbesondere, daß, wenn der Programmschritt-Anzeiger an einen **g** **NOP**-Befehl (No OPeration, Leerbefehl) kommt, keine Operation ausgeführt wird.

Beachten Sie weiter, daß **STO** **9** und **RCL** **9** nicht kombiniert werden. Dies kann Ihnen als Gedächtnisstütze dafür dienen, daß der HP-65 das Register R₉ zur Speicherung von Zwischenergebnissen bei der Durchführung von trigonometrischen Funktionen, Rechtwinklig/Polar-Umwandlungen und numerischen Vergleichen verwendet.

60 Programmierung

ANZEIGE DES SPEICHERENDES

Wenn Sie weiterhin die Taste **SST** drücken, kann es vorkommen, daß Sie die Speicherzeile 100 erreichen und in der Anzeige zwei Minuszeichen auftauchen:

— 35 01 —

Damit soll angezeigt werden, daß das Speicherende erreicht ist. Nach nochmaligem Drücken von **SST** steht der Programmschritt-Anzeiger wieder am Anfang des Speichers (00 00).

ANZEIGE FÜR VOLLEN PROGRAMMSPEICHER

Immer wenn die letzte Speicherzeile (Zeile 100) etwas anderes enthält als **NOP**, wird im W/PRGM-Modus jeder Code mit einem Minuszeichen auf der rechten Seite angezeigt. Daran können Sie stets erkennen, ob der Programmspeicher bereits voll belegt ist. Wurde beispielsweise der Programmschritt-Anzeiger auf einen **RTN**-Befehl in Speichermitte gesetzt und war der Speicher bereits vollständig belegt, so erhält man folgende Anzeige:

24 —

LÖSCHEN DES PROGRAMMSPEICHERS

Mit der Tastenfolge **f** **PRGM** können Sie im W/PRGM-Modus den Programmspeicher löschen. Dies ist einer der Befehle, die nicht in den Speicher geschrieben werden können. Es ist unbedingt erforderlich, den Programmspeicher zu löschen, bevor Sie eine oder mehrere der Tasten **A** bis **E** neu definieren. Anderenfalls schieben Sie die vorherigen Programme im Speicher weiter, was dazu führen kann, daß Sie nachher mit einer der Tasten **A** bis **E** zwei identisch markierte Programme kontrollieren.

Zum Löschen des Programmspeichers schalten Sie in Stellung W/PRGM und drücken Sie:

f **PRGM**

Damit werden sämtliche 100 Speicherschritte mit **9** **NOP** belegt und der Programmschritt-Anzeiger an den Speicheranfang gesetzt (Anzeige 00 00).

EIN EIGENES PROGRAMM

Jetzt, da Sie schon etwas mehr über den Programmspeicher des HP-65 erfahren haben, können Sie ein anderes Programm erstellen.

Das Programm soll das Volumen einer Kugel unter Verwendung der folgenden Formel berechnen: $\text{Volumen} = r^3 \times \pi \times 4/3$. Alles, was Sie zu tun haben, ist, den Radius (r) einzutasten und **A** zu drücken. Folgen Sie für das Eintasten des Programms den nachstehenden Anweisungen:

1. Schalten Sie in Stellung W/PRGM.
2. Drücken Sie **f** **PRGM**, um den Programmspeicher zu löschen und den Programmschritt-Anzeiger an den Speicheranfang zu setzen.
3. Drücken Sie die Tasten in der angegebenen Reihenfolge. Nehmen Sie sich dabei die Zeit und identifizieren Sie jede der Tasten nach ihrem Code.

Tastencode	Taste
23	LBL
11	A
03	3
35	9
05	y^x
35	9
02	π
71	×
04	4
71	×
03	3
81	÷
24	RTN

Bemerkung

Hier beginnt die Ausführung nach Drücken von **A**.

Berechnet r^3

Berechnet $r^3 \times \pi$

Berechnet $r^3 \times \pi \times 4/3$

Bezeichnet das Programmende

62 Programmierung

Wenn Ihnen bei der Eingabe ein Fehler unterläuft, löschen Sie den Programmspeicher und beginnen Sie von neuem. Sie werden noch erfahren, wie Fehler auf andere Weise zu beheben und Programme zu korrigieren sind.

AUSFÜHREN DES PROGRAMMS

Um das Programm auszuführen, schalten Sie den W/PRGM-RUN-Schalter in Stellung RUN. Jetzt können Sie das Volumen einer Kugel mit Radius 10 berechnen.

Drücken Sie	Anzeige
10 A	4188.79 Kugelvolumen

Mit dem Drücken von **A** begann der Programmschritt-Anzeiger von seinem augenblicklichen Standort aus mit der Suche nach **LBL A**. Nach Auffinden der entsprechenden Marke begann von dort aus die Ausführung der nachfolgenden Programmschritte. Wäre im Programmspeicher keine Marke A zu finden gewesen, hätte der Rechner am Speicheranfang mit der Ausführung begonnen. Falls Sie gerade das zuletzt genannte Beispiel gerechnet haben, schalten Sie in Stellung W/PRGM. Als Anzeige erhalten Sie den Code der zuletzt ausgeführten Anweisung: .

Der **RTN**-Befehl am Ende des Programms hält den Programmschritt-Anzeiger an, stoppt die weitere Ausführung von Programmschritten und gibt die Kontrolle an das Tastenfeld zurück.

Wenn Sie das Gesamtvolumen von fünf Kugeln mit Radius 10 berechnen wollen, multiplizieren Sie das Ergebnis einfach mit 5. Diese Rechnungen haben auf das Programm keinen Einfluß. Schalten Sie zurück in den RUN-Modus und versuchen Sie es.

Drücken Sie	Anzeige
10 A	4188.79 Volumen einer Kugel
5 x	20943.95 Volumen von 5 Kugeln

Schalten Sie jetzt wieder in Stellung W/PRGM. Als Anzeige erhalten Sie: 71.

Dies ist der Tastencode für Multiplikation. Obwohl der Programmschritt-Anzeiger an einem **RTN** steht (Programmende), zeigt die Anzeige den Code der zuletzt ausgeführten Operation.

MAGNETKARTEN

Zeichnen Sie Ihr Programm jetzt auf eine Magnetkarte auf, indem Sie:

1. eine unbeschriebene, ungeschützte Magnetkarte auswählen;
2. in die Stellung W/PRGM schalten;
3. die Magnetkarte in den unteren rechten Schlitz des Rechners einführen, wie Sie das auch für das Einlesen eines Programms schon getan haben.

Die Stellung des W/PRGM-RUN-Schalters ist von großer Bedeutung, wenn Sie Programme aufzeichnen oder von bereits programmierten Magnetkarten in den Rechner einlesen.

Zum Aufzeichnen Ihres eigenen Programms muß der Schalter in Stellung W/PRGM (Write PRoGraM = Programm schreiben) stehen.

Zum Einlesen beschriebener Magnetkarten muß der Wahlschalter in Stellung RUN (RUN = Programm ausführen) stehen.

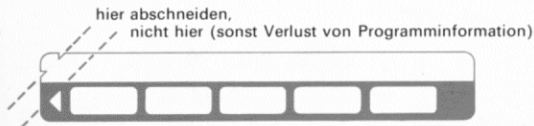
LESE/SCHREIB-OPERATIONEN

Beim Lesen oder Schreiben einer Magnetkarte werden stets alle 100 Programmschritte übertragen. Dabei werden die Inhalte der Register nicht verändert, so daß Sie mit Daten weiterrechnen können, die von dem vorherigen Programm entwickelt wurden.

Tritt beim Lesen einer Magnetkarte ein Fehler auf, wird der gesamte Programmspeicher gelöscht (d.h. mit **9** **NOP** besetzt) und die Anzeige blinkt. Den gleichen Effekt hat der Versuch, eine unbeschriebene Karte einzulesen.

SCHÜTZEN EINER MAGNETKARTE

Wenn Sie eine Magnetkarte gegen unbeabsichtigtes Löschen oder Überschreiben schützen wollen, entfernen Sie mit einer Schere den markierten Eckenabschnitt, wie es Ihnen die nachstehende Zeichnung zeigt.



Eine weitere Schutzmaßnahme können Sie treffen, indem Sie das Programm auch auf der zweiten Spur der Magnetkarte aufzeichnen. Dazu müssen Sie die Karte mit dem anderen Ende voraus in den Rechner einführen. Wenn Sie jetzt versehentlich Ihre Programminformation löschen, können Sie die Karte immer noch in Gegenrichtung lesen. Für die bleibende Speicherung Ihrer Programme sollten Sie allerdings nur die eine vorgesehene Spur der Magnetkarten verwenden. Dafür gibt es eine Reihe von Gründen:

1. Die Karte kann für das zweite Programm nicht ohne weiteres beschriftet werden.
2. Beim eventuellen Schützen der zweiten Spur muß mit größter Vorsicht vorgegangen werden. Keinesfalls dürfen Sie mehr abschneiden, als an der gegenüberliegenden Kartenecke für die Spur 1 vorgesehen ist, da Sie sonst Programminformation verlieren.
3. Die Motor-Antriebsrolle des Kartenlesers liegt auf dieser zweiten Spur. Das kann dazu führen, daß nach längerem Gebrauch beim Lesen dieser Spur Fehler auftreten.

BESCHRIFTEN DER MAGNETKARTEN

Die nicht-magnetische Seite der Programmkarten können Sie mit beliebigem Schreibgerät beschriften, solange Sie die Karte dabei nicht beschädigen. Es muß vermieden werden, daß durch das Einprägen bei zu großem Schreibdruck die Oberfläche der Magnetschicht wellig wird. Daher sollten Sie von einer Beschriftung der Karten mit Schreibmaschine absehen.

Es ist zweckmäßig, einen Programmnamen in die obere Zeile der Karte zu schreiben. In die darunterliegenden Kästchen können Sie Symbole setzen, die die Funktion der entsprechenden Tasten **A** bis **E** erklären.

Für die bleibende Beschriftung der Magnetkarten hat sich Tusche bewährt, wobei Sie die Karte vorher von Staub und Fett reinigen sollten. Verwenden Sie dazu nach Möglichkeit keine chemischen Lösungsmittel (wie z.B. Benzin, Alkohol usw.), da dadurch die Farbe des Kartenaufdrucks aufgelöst werden kann.

KORREKTUR EINES PROGRAMMS

Sie können die Programme zu Ihrem HP-65 auf einfache Weise abändern bzw. Fehler beseitigen, wenn Sie dafür die verschiedenen Korrekturfunktionen verwenden, über die der Rechner verfügt. Diese Funktionen ermöglichen es, an beliebiger Stelle des Programms einzelne Programmschritte einzufügen oder zu löschen.

POSITIONIEREN DES PROGRAMMSCHRITT-ANZEIGERS

Bevor Sie ein Programm korrigieren können, müssen Sie zuvor den Programmschritt-Anzeiger an die entsprechende Programmstelle setzen. Sie haben bereits eine Möglichkeit kennengelernt, mit der Sie das erreichen können. Mit Hilfe der Taste **SST** können Sie den Programmschritt-Anzeiger schrittweiseiterrücken. Diese Möglichkeit ist allerdings wenig bequem, wenn die abzuändernde Programmstelle nahe dem Speicherende liegt.

Es gibt einen wesentlich einfacheren Weg.

Sie können den Programmschritt-Anzeiger auf eine beliebige im Programm vorkommende Marke **LBL** (LaBeL) setzen, indem Sie im RUN-Modus **GTO** (Go TO) und dann die entsprechende Taste (**A** bis **E**)* drücken.

* Sie können den Programmschritt-Anzeiger ebenso auf die Marken 1 bis 9 vorrücken, indem Sie **GTO** (**1** bis **9**) drücken. Die Marken werden auf Seite 72 besprochen.

66 Programmierung

Der Programmschritt-Anzeiger sucht daraufhin den Speicher nach der entsprechenden Marke ab und hält an, sobald er diese Stelle erreicht hat. Wenn Sie zum Beispiel **GTO C** drücken, sucht der Programmschritt-Anzeiger die Marke C (**LBL C**). Wenn er sie gefunden hat, hält er bei dem Programmschritt an, der die Information **C** enthält. Kann eine entsprechende Marke nicht gefunden werden, springt der Programmschritt-Anzeiger an den Speicheranfang und stoppt. Sie können jetzt in Stellung W/PRGM schalten und den Programmschritt-Anzeiger aus der Position des **C** mit **SST** an die Korrekturstelle vorrücken. Auf diese Weise haben Sie einen langen Abschnitt des Programms übersprungen.

Für das Zurücksetzen des Programmschritt-Anzeigers an den Speicheranfang haben Sie zwei Möglichkeiten:

1. Sie drücken **SST**, bis Sie den Speicher zyklisch ganz durchlaufen haben und wieder am Speicheranfang (**00 00**) ankommen.
2. Sie schalten in Stellung RUN und drücken **RTN**.

Für welche der beiden Möglichkeiten Sie sich entscheiden, wird davon abhängen, an welcher Stelle des Programmspeichers Sie sich gerade befinden. In der Regel ist die Verwendung von **RTN** im RUN-Modus bequemer. (Beachten Sie, daß **RTN** im RUN-Modus anders wirkt als im W/PRGM-Modus.)

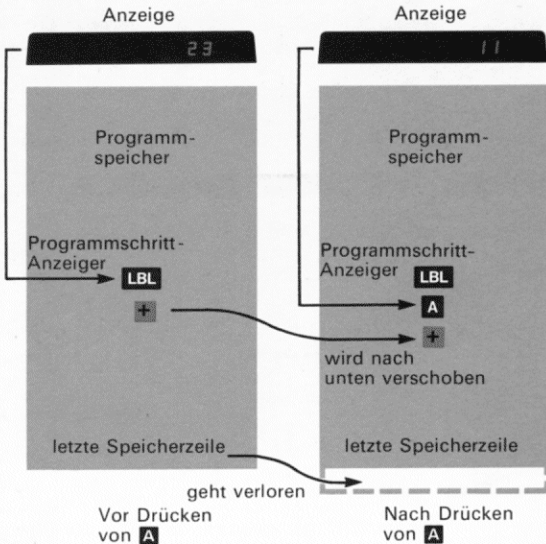
EINFÜGEN VON PROGRAMMSCHRITTEN

Auch wenn Sie es gar nicht bemerkt haben, so haben Sie doch schon erfahren, wie Programmschritte eingefügt werden können. Immer wenn Sie bisher ein Programm geschrieben und eine der Tasten im W/PRGM-Modus gedrückt haben, wurde der Programmschritt zwischen den angezeigten und den nächstfolgenden in den Speicher geschrieben. Der Programmschritt-Anzeiger wurde dann weitergerückt, um den eingefügten Programmschritt anzuzeigen.

Zusammengefaßt lautet die Anweisung für das Einfügen von Programmschritten:

1. Setzen Sie den Programmschritt-Anzeiger so, daß der Code des Programmschrittes angezeigt wird, hinter dem die neuen Schritte einzufügen sind.
2. Drücken Sie die Tasten für die einzufügenden Programmschritte. Dabei werden die nachfolgenden Programmteile im Speicher nach unten geschoben. (Es werden keine Speicherzeilen «überschrieben»!)

Wie Sie aus der nachstehenden Zeichnung ersehen, geht beim Einfügen eines Programmschrittes der Inhalt der letzten Programmspeicherzeile verloren. (Dies ist natürlich nur dann von Bedeutung, wenn ein Minuszeichen in der Anzeige darauf hinweist, daß der Speicher ganz belegt ist.)



Für den zweiten Teil eines kombinierten Codes allein kann keine Einfüge-Operation ausgeführt werden, da hier der zweite Tastenbefehl in der gleichen Zeile gespeichert wird.

68 Programmierung

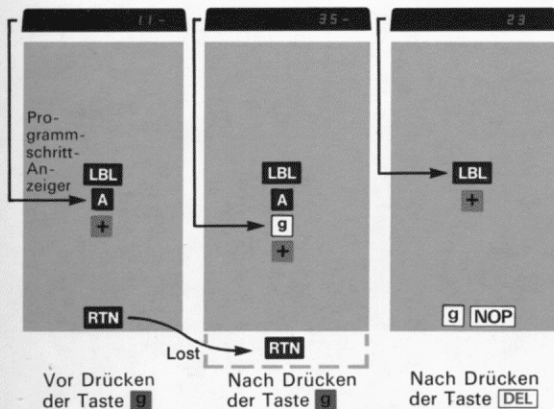
Ebenfalls können Einfüge-Operationen nicht ausgeführt werden, wenn der Programmschritt-Anzeiger am Speicherende steht. In einem solchen Fall werden zwar die zu den gedrückten Tasten gehörenden Codes angezeigt, nicht aber in den Speicher geschrieben.

LÖSCHEN VON PROGRAMMSCHRITTEN

Sie können jederzeit einzelne Schritte eines Programms löschen, indem Sie nach den folgenden Anweisungen verfahren:

1. Setzen Sie den Programmschritt-Anzeiger so, daß der zu löschende Programmbefehl angezeigt wird.
2. Drücken Sie **g** **DEL** (DELeTe = Löschen) im W/PRGM-Modus. Der Programmschritt wird daraufhin entfernt und der Programmschritt-Anzeiger auf den davorliegenden Schritt zurückgesetzt.

Die nachstehenden Zeichnungen veranschaulichen die Wirkungsweise der **g** **DEL**-Operation auf den Programmspeicher.



Wie Sie aus der obenstehenden Zeichnung erkennen, wird der Tastenbefehl **g** zuerst als neuer Bestandteil eines Programms

in den Speicher geschrieben, worauf alle nachfolgenden Schritte um eine Speicherzeile nach unten verschoben werden.

Da in dem skizzierten Fall der Speicher ganz belegt ist (Minuszeichen in der Anzeige), geht der letzte Programmschritt dabei verloren, wie dies bei jeder Einfüge-Operation der Fall ist. Sobald aber jetzt der Folgeschritt **DEL** eingetastet wird, erkennt der Rechner die vollständige Anweisung und entfernt daraufhin sowohl den Tastenbefehl **9** als auch den davorliegenden unerwünschten Programmschritt.

Der Programmschritt-Anzeiger rückt zwei Schritte zurück, und die beiden letzten Speicherpositionen werden mit **9** **NOP** belegt. Vergessen Sie nicht, den letzten Schritt Ihres Programms erneut einzutasten. (Für Programme, die kürzer sind als 100 Schritte, sind die Vorgänge am Speicherende natürlich unbedeutend.)

LÖSCHEN AUFEINANDERFOLGENDER SCHRITTE

Zum Löschen einer Folge von Programmschritten ist der Programmschritt-Anzeiger auf den letzten zu entfernenden Schritt zu stellen. Mit jedem Drücken von **9** **DEL** wird der Programmschritt-Anzeiger zurückgerückt und der nächste zu löschende Schritt angezeigt.

LÖSCHEN DER LETZTEN SPEICHERZEILE

Wenn der Programmschritt-Anzeiger am Speicherende steht und **9** **DEL** gedrückt wird, werden zwei Speicherpositionen gelöscht: Speicherzeile 100 und 99. Denken Sie in einer solchen Situation daran, den zusätzlich entfernten 99. Schritt erneut einzutasten!

ZURÜCKRÜCKEN DES PROGRAMMSCHRITT-ANZEIGERS

Wenn Sie sich unter Verwendung von **SST** schrittweise durch den Speicher tasten, kann es vorkommen, daß Sie dabei über das Ziel hinauspringen. Sie können dann mit **9** **DEL** den Programmschritt-Anzeiger zurückrücken, wobei Sie allerdings

70 Programmierung

die dazwischenliegenden Programmschritte löschen. Nach Einfügen oder Entfernen von Programmschritten an der gewünschten Stelle müssen Sie die beim Zurückrücken gelöschten Schritte wieder eintasten. In vielen Fällen ist das bequemer, als den Programmschritt-Anzeiger mit anderen Mitteln erneut zu positionieren.

KORRIGIEREN EINES PROGRAMMS

Nachdem Sie nun mit den verschiedenen Korrekturoperationen vertraut sind, wollen wir das Gelernte in der Praxis anwenden.

Das Programm zur Berechnung des Kugelvolumens soll so abgeändert werden, daß es die Kugeloberfläche ($r^2 \times \pi \times 4$) berechnet. Die beiden Programme ähneln sich sehr. Anderenfalls wäre es nicht so einfach möglich, sie ineinander umzuwandeln. Sehen Sie sich die Nebeneinanderstellung beider Programme an.

Kugelvolumen	Kugeloberfläche
LBL } Beginn des	LBL } Beginn des
A } Programms	A } Programms
3 } Berechnet r^3	2 } Berechnet r^2 *
g }	g }
y^x }	y^x }
g } Multiplikation mit π	g } Multiplikation mit π
π }	π }
x }	x }
4 } Multiplikation mit 4	4 } Multiplikation mit 4
x }	x }
3 } Division durch 3	RTN Ende des Programms
\div }	
RTN Ende des Programms	

Wie Sie sehen, müssen nicht viele Programmschritte geändert werden. Tasten Sie jetzt das Programm zur Berechnung des

* Das gleiche Resultat könnte auch mit der Tastenfolge **f** **y^x** oder **ENTER** **x** berechnet werden, wobei ein Schritt eingespart würde. Es wurde davon abgesehen, um die Aufgabe nicht komplizierter zu gestalten.

Kugelvolumens ein, sofern Sie das nicht schon getan haben. Folgen Sie dazu den nachstehenden Anweisungen:

1. Schalten Sie den Rechner in den W/PRGM-Modus.
2. Drücken Sie **f** **PRGM** zum Löschen des Programmspeichers.
3. Geben Sie die in der linken Spalte angegebene Tastenfolge ein.
4. Schalten Sie zurück in den RUN-Modus.

Testen Sie Ihr Programm anhand des nachstehenden Beispiels, bevor Sie es abändern.

Beispiel : Berechnen Sie das Volumen einer Kugel mit Radius 25.

Drücken Sie	Anzeige
25 A	65449.85

Um das Kugelvolumen-Programm derart zu modifizieren, daß die Kugeloberfläche berechnet wird, müssen folgende Änderungen vorgenommen werden:

Kugelvolumen	Kugeloberfläche
LBL	LBL
A	A
3 ← Löschen Sie diesen Schritt	2 ← Fügen Sie diesen Schritt ein
g	g
y^x	y^x
g	g
π	π
x	x
4	4
x	x
3 } Löschen Sie diese beiden Schritte	RTN
÷ }	
RTN	

Und so müssen Sie verfahren:

1. Schalten Sie in Stellung RUN.
2. Drücken Sie **GTO** **A**, um den Programmschritt-Anzeiger auf **LBL** **A** zu setzen.

72 Programmierung

3. Schalten Sie wieder in Stellung W/PRGM.
4. Drücken Sie die Taste **SST** einmal, um den Programmschritt-Anzeiger auf den Schritt zu setzen, der entfernt werden soll. Die Anzeige sollte den Code 03 enthalten.
5. Drücken Sie **g** **DEL** zum Löschen des unerwünschten Programmschrittes. In der Anzeige erscheint 11, der Code für **A**.
6. Drücken Sie **2**, um diesen Schritt einzufügen.
7. Drücken Sie neunmal **SST**. Der Programmschritt-Anzeiger steht jetzt auf dem zweiten der beiden aufeinanderfolgenden Schritte, die zu entfernen sind. Die Anzeige sollte den Code 81 enthalten.
8. Drücken Sie **g** **DEL** und löschen Sie damit den Schritt \div . Der Programmschritt-Anzeiger rückt eine Zeile zurück; neue Anzeige: 03.
9. Drücken Sie nochmals **g** **DEL** und entfernen Sie damit den Schritt **3**. Die Anzeige sollte den Tastencode 71 enthalten.
10. Schalten Sie jetzt zurück in Stellung RUN. Das neue Programm kann ausgeführt werden.

Tasten Sie einen Wert für den Radius r ein und führen Sie das Programm aus, indem Sie **A** drücken.

Beispiel: Berechnen Sie die Oberfläche einer Kugel mit $r = 25$.

Drücken Sie	Anzeige
25 A	7853.98

Sie können jetzt zu Übungszwecken das vorliegende Programm wieder so abändern, daß das Volumen einer Kugel berechnet wird.

PROGRAMMSPRÜNGE

Normalerweise führt der HP-65 die Programme sequentiell aus, d.h. er arbeitet das Programm schrittweise in der Reihenfolge ab, in der die Programmschritte im Speicher stehen. Er besitzt darüber hinaus aber auch die Fähigkeit, während der Ausführung eines Programms zu einem markierten Programmteil

(**LBL**, ...) zu springen und von da ab in der Ausführung fortzufahren.

MARKEN

Eine Marke besteht aus der Tastenfolge **LBL**, gefolgt von einer der Tasten **0** bis **9** oder einer der Programmkontrolltasten **A** bis **E**. Diese 15 Marken können beliebig innerhalb eines Programms verwendet werden. Dabei ist allerdings zu beachten, daß nur solche Programmteile vom Tastenfeld direkt aufgerufen werden können, die mit den Programmkontrolltasten **A** bis **E** markiert sind.

DIREKTE (UNBEDINGTE) SPRÜNGE

Ein direkter Sprungbefehl innerhalb des Programms besteht aus der Tastenfolge **GTO** (Go TO = springe nach...), gefolgt von einer der Tasten **0** bis **9** oder **A** bis **E**. Er muß sich jeweils auf eine entsprechende Marke irgendwo innerhalb des Programms beziehen. Ist eine entsprechende Marke nicht vorhanden, setzt der Rechner die Ausführung des Programms am Speicheranfang fort. Wenn der Rechner einen solchen direkten Sprung ausführt, sucht der Programmschritt-Anzeiger von der Position des **GTO** aus den Speicher auf die entsprechende Marke ab und setzt die Programmausführung fort, sobald er diese gefunden hat. So läßt zum Beispiel **GTO 3** den Programmschritt-Anzeiger nach **LBL 3** springen, worauf von dieser Stelle aus die Ausführung der Programmbefehle weiterläuft. Beachten Sie, daß **GTO 3** vom Tastenfeld aus die gleiche Wirkung hat, nur daß in diesem Fall die Programmausführung nicht fortgesetzt wird.

ERSTELLEN EINES PROGRAMMS MIT EINEM DIREKTEM SPRUNG

Direkte Sprünge werden häufig dann angewandt, wenn mehrere Programmteile gemeinsame Tastenfolgen beinhalten. Wir wollen ein solches Programm schreiben, um dies zu verdeut-

74 Programmierung

lichen. Nehmen Sie an, Sie sollen Programme zur Berechnung von zwei einander ähnlichen Formeln erstellen:

$$y = \frac{\sin x}{3 (\sin x)^2 + 2}$$

$$y = \frac{\cos x}{3 (\cos x)^2 + 2}$$

Sie können auf einfache Weise zwei getrennte Programme schreiben und sie den Programmkontrolltasten **A** und **B** zuordnen.

Tasten	Bemerkungen
LBL } A } f } SIN } ENTER ↑ } ENTER ↑ } × } 3 } × } 2 } + } ÷ } RTN	<p>Hier beginnt die Ausführung</p> <p>Berechnet $\sin x$</p> <p>Speichert Wert des Zählers $(\sin x)^2$</p> <p>$3 (\sin x)^2$</p> <p>$3 (\sin x)^2 + 2$</p> <p>Resultat</p>

Tasten	Bemerkungen
LBL } A } f } COS } ENTER ↑ } ENTER ↑ } × } 3 } × } 2 } + } ÷ } RTN	<p>Hier beginnt die Ausführung</p> <p>Berechnet $\cos x$</p> <p>Speichert Wert des Zählers $(\cos x)^2$</p> <p>$3 (\cos x)^2$</p> <p>$3 (\cos x)^2 + 2$</p> <p>Resultat</p>

Wie Sie erkennen, sind die letzten neun Schritte beider Programme gleich. Daher wollen wir diesen Teil zu einem Programmabschnitt zusammenfassen und mit **LBL C** markieren. Sowohl **LBL A**-Programm als auch **LBL B**-Programm können dann zu diesem nur einmal vorhandenen gemeinsamen Programmteil springen. In der **C**-Routine wird der Ausdruck

$$\frac{a}{3a^2 + 2} \text{ berechnet.}$$

Für das eine Programm gilt $a = \sin x$, für das andere $a = \cos x$. Schalten Sie den Rechner in Stellung **W/PRGM**, drücken Sie **f PRGM** und tasten Sie jetzt das folgende Programm ein:

Tasten	Bemerkungen
LBL	Beginn
C	
ENTER↑	Speichern von a
ENTER↑	
×	a^2
3	
×	$3a^2$
2	
+	$3a^2 + 2$
÷	Resultat
RTN	Ende

Tasten Sie jetzt die beiden **A**- und **B**-Programme ein, die inzwischen wesentlich gekürzt sind:

Tasten	Bemerkungen
LBL	Beginn
A	
f	Berechnet $\sin x$
SIN	
GTO	Sprung nach LBL C
C	

Tasten	Bemerkungen
<div> <div>LBL</div> <div>B</div> <div>f</div> <div>COS</div> <div>GTO</div> <div>C</div> </div>	Beginn Berechnet $\cos x$ Sprung nach LBL C

Beachten Sie, daß die beiden Programme nicht mit **RTN** enden, da sie direkt nach **LBL C** springen und die Programmausführung dort weiterläuft. Außerdem ist Ihnen vielleicht aufgefallen, daß wir zuerst das Programm **C** und dann erst die Programme **A** und **B** eingetastet haben. Es ist nicht von Bedeutung, in welcher Reihenfolge die Programme im Speicher stehen.

Schalten Sie den Rechner jetzt in den RUN-Modus und führen Sie die neuen Programme aus.

Beispiel: Berechnen Sie $\frac{\sin x}{3(\sin x)^2 + 2}$ und $\frac{\cos x}{3(\cos x)^2 + 2}$ für $x = 60^\circ$.

Drücken Sie

9 **DEG**

Anzeige

0.00

Bemerkungen

Setzt Winkel-Modus «Grad» falls nicht bereits gesetzt

60 **A**

0.20

Ergebnis

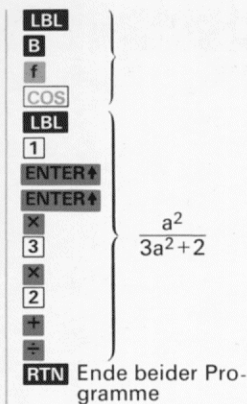
60 **B**

0.18

Ergebnis

Sie können das Programm noch weiter verbessern (und damit Speicherplatz einsparen), indem Sie die Marken wie folgt umsetzen:

Tasten	Bemerkungen
<div> <div>LBL</div> <div>A</div> <div>f</div> <div>SIN</div> <div>GTO</div> <div>1</div> </div>	Dieses Programm berechnet $\sin x$ und springt dann nach Marke 1



Dieser Teil berechnet $\cos x$ und setzt dann die Ausführung über **LBL 1** hinaus fort

Beginn des gemeinsamen Programmabschnitts

Beachten Sie als erstes, daß **LBL C** durch **LBL 1** ersetzt wurde. Da wir nicht beabsichtigen, diesen Programmteil vom Tastenfeld aufzurufen, ist es nicht nötig, eine der wertvollen Programmkontrolltasten dafür zu verwenden. Zum zweiten haben wir zwei Programmschritte dadurch einsparen können, daß wir **LBL B** unmittelbar vor **LBL 1** (vorher **LBL C**) gesetzt haben. Auf diese Weise ist ein Sprung von **LBL B** nach **LBL 1** nicht mehr nötig.

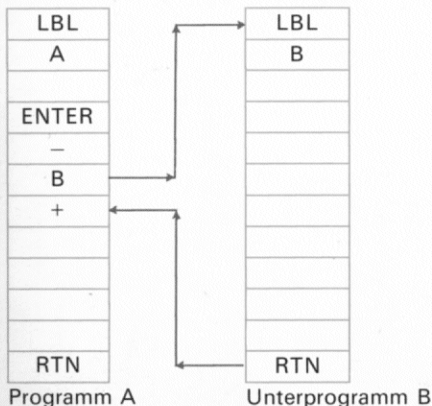
SPRUNG ZU UNTERPROGRAMMEN

Eine zweite Möglichkeit, die Ausführung eines Programms zu verzweigen, besteht in der Verwendung von «Unterprogrammen» (auch Routinen genannt). Falls ein Programm eine Reihe von Einzelschritten häufig durchläuft oder dieser Programmteil mehreren Programmen gemeinsam ist, kann ein einzelnes Unterprogramm erstellt werden, das diese Schritte beinhaltet.

Ebenso wie Sie die Programmkontrolltasten **A** bis **E** zum Aufrufen der entsprechenden Programmteile zwischen den zugeordneten **LBL** und **RTN** verwenden, so kann auch der Rechner davon Gebrauch machen. Wenn dann ein **RTN**-Befehl

78 Programmierung

erreicht wird, wird das Programm nicht angehalten, sondern die Ausführung des Programms geht ab dem Schritt weiter, der dem ursprünglichen Sprungbefehl folgt.



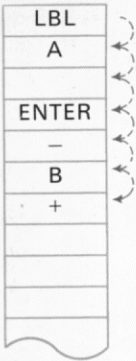
Wie Sie sicher bemerken, ist eine solche Routine oder ein solches Unterprogramm ein Programm wie jedes andere auch. Der Unterschied besteht hier in der Anwendung. Wenn Sie in dem vorstehend skizzierten Beispiel **B** drücken, wird das zugeordnete Programm ausgeführt, und der Rechner hält bei dem **RTN** an. Drücken Sie dagegen **A**, arbeitet der Rechner das Programm unter **LBL A** schrittweise ab, bis er an den Befehl **B** kommt. Hier springt er nach **LBL B** und setzt die Ausführung dieser Programmschritte fort. Beim zugehörigen **RTN** angelangt, springt er jetzt aber zurück zum Programm **A** und fährt mit der Ausführung ab dem Programmschritt fort, der dem **B** folgt.

Mit anderen Worten ausgedrückt, stellt die Taste **B** einen weiteren Programmschritt des **A**-Programms dar. Die Ausführung ist die gleiche, wie wenn die Tastenfolge vom Tastenfeld aus gedrückt würde. Beachten Sie bitte in diesem Zusammenhang, daß nur die Marken **A** bis **E** Unterprogramme anführen können, nicht aber die Marken **0** bis **9**.

Sie sollten daran denken, daß der Aufruf von Unterprogrammen häufig Daten im Stack umordnet oder verändert. Diese Möglichkeit müssen Sie bei der Erstellung eines entsprechenden Programms beachten und gegebenenfalls Werte zur späteren Wiederverwendung abspeichern.

ZWEITER PROGRAMMSCHRITT-ANZEIGER *

An irgendeiner Stelle muß sich der Rechner merken, wohin er nach Ausführung eines Unterprogramms zurückspringen muß. Dazu verwendet er einen zweiten Programmschritt-Anzeiger. Wenn im vorstehenden Beispiel **A** gedrückt wird, führt er Schritt für Schritt aus, bis der Haupt-Programmschritt-Anzeiger **B** erreicht. Hier hält der Programmschritt-Anzeiger an und markiert seine Stellung unmittelbar hinter dem **B**-Befehl.

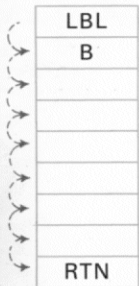


Der Haupt-Programmschritt-Anzeiger hält an und markiert seine Position nach Ausführung des Unterprogrammaufrufs **B**.

Zwischenzeitlich wird ein zweiter Programmschritt-Anzeiger aktiviert. Am Speicheranfang beginnend, sucht er nach **LBL B**. Wenn er die Marke gefunden hat, führt er die unter **LBL B** stehenden Programmschritte des Unterprogramms aus.

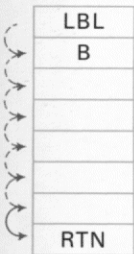
* Für fortgeschrittene Programmierung ist es von großem Wert, wenn Sie die Wirkungsweise der Programmschritt-Anzeiger innerhalb des Rechners verstehen. Dieses Gebiet wird in Anhang B diskutiert.

80 Programmierung

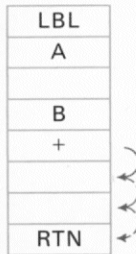


Der zweite Programmschritt-Anzeiger führt das Unterprogramm **B** aus.

Mit Erreichen des **RTN** im Unterprogramm wird der zweite Programmschritt-Anzeiger unwirksam, und der Haupt-Programmschritt-Anzeiger wird wieder aktiviert. Damit geht die Programmausführung im Hauptprogramm weiter.



Zweiter Programmschritt-Anzeiger wird nach Ausführung des **RTN** stillgelegt.



Haupt-Programmschritt-Anzeiger wird wieder aktiviert. Die Programmausführung geht bei **+** weiter.

ERSTELLEN EINES PROGRAMMS MIT UNTERPROGRAMM

Um die Oberfläche und das Volumen einer Kugel zu berechnen, verwenden wir sinnvollerweise ein Programm mit Unterprogramm. Die Formeln für die beiden Berechnungen sind:

$$\text{Oberfläche} = r^2 \times \pi \times 4 \quad \text{und} \quad \text{Volumen} = \frac{r^3 \times \pi \times 4}{3}$$

Die Volumen-Gleichung kann leicht als Funktion der Oberfläche ausgedrückt werden:

$$\text{Volumen} = \frac{r \times (\text{Fläche})}{3}$$

Diese letzte Formel wollen wir bei der Erstellung des Programms nutzen. Das mit **A** markierte Programm berechnet die Kugeloberfläche. Schalten Sie in den W/PRGM-Modus, drücken Sie **f** **PRGM** zum Löschen des Programmspeichers und geben Sie dann die nachstehende Tastenfolge ein.

Taste	Bemerkungen
LBL }	Programmbeginn
A }	
STO 1	Speichern von r für spätere Verwendung
f1 }	
√x }	Berechnet r^2
9 }	
π }	Aufruf von π
x	
4	
x	
RTN	Kugeloberfläche Programmende

Schalten Sie jetzt um in den RUN-Modus und testen Sie das Programm.

Beispiel: Berechnen Sie die Oberfläche einer Kugel mit Radius $r = 15$.

Drücken Sie

15 **A**

Anzeige

2827.43 Kugeloberfläche

82 Programmierung

Unter Verwendung dieses Programms wollen wir jetzt das Volumen dieser Kugel berechnen.

Drücken Sie

15 **A**

RCL **1**

x

3

÷

Anzeige

2827.43 Kugeloberfläche

15.00 Rückruf des Wertes für r

42411.50 $r \times \text{Fläche}$

3.00

14137.17 Kugelvolumen

Um diese Tastenfolge zu einem eigenständigen Programm zu machen, brauchen wir nur ein **LBL** vor und ein **RTN** hinter die Schrittfolge zu setzen.

Taste	Bemerkungen
LBL } B } A } RCL 1 x 3 ÷ RTN	Programmbeginn Aufruf für Routine A Rückruf des Wertes für den Radius r $r \times \text{Fläche}$ Kugelvolumen Programmende

Beachten Sie, daß wir den Wert für den Radius nicht erneut einzugeben brauchten, sondern ihn aus Register 1 zurückrufen konnten. Schalten Sie um in Stellung W/PRGM und tasten Sie das neue Programm ein. Drücken Sie jetzt *nicht* **f** **PRGM**, denn wir wollen das Programm A ja im Speicher belassen. Wir können jetzt beide Programme einsetzen.

Beispiel: Berechnen Sie Volumen und Oberfläche einer Kugel mit Radius $r = 20$.

Drücken Sie

20 **A**

20 **B**

Anzeige

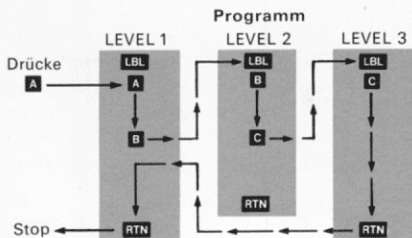
5026.55 Oberfläche der Kugel

33510.32 Volumen der Kugel

Jetzt berechnet der Rechner das Volumen der Kugel nach genau der gleichen Methode, die Sie im vorangegangenen Beispiel angewandt haben.

UNTERPROGRAMME IN ZWEITER EBENE

Ein Unterprogramm kann nicht seinerseits ein Unterprogramm aufrufen. Der Grund dafür liegt im Fehlen eines dritten Programmschritt-Anzeigers, der in einem solchen Fall die Steuerung des Programmablaufs in der zweiten Routine übernehmen müsste. Wenn Sie den Versuch trotzdem unternehmen, werden Sie feststellen, daß nach Ablauf des sekundären Unterprogramms die Kontrolle an das Hauptprogramm und nicht an die erste Routine zurückgeht.



VERGLEICHS-OPERATIONEN, «BEDINGTE» SPRÜNGE

Der HP-65 verfügt über neun Tastenbefehle, die ihm die Möglichkeit geben, innerhalb eines Programms Entscheidungen zu treffen. Mit Hilfe solcher Vergleichs- und Test-Operationen kann der Ablauf des Programms verzweigt werden. Die Wirkungsweise dieser neun Operationen gleicht sich im wesentlichen.

Wenn die zu prüfende Bedingung *erfüllt* ist, fährt der Rechner mit der schrittweisen Ausführung des Programms fort, wogegen er die beiden darauffolgenden Programmschritte überspringt, wenn die Bedingung *nicht erfüllt* ist. Häufig bestehen diese beiden Schritte aus einer Sprunganweisung (**GTO**, ...), womit eine beliebige Verzweigung des Programms möglich ist. Bisweilen kann man sich auf diese Weise die Anweisungen sparen, die einen Sprung zu den zwei Programmschritten veranlassen würden. Für beide genannten Möglichkeiten werden Sie im weiteren Text Beispiele finden.

NUMERISCHE VERGLEICHS-OPERATIONEN

Vier Testfunktionen vergleichen jeweils die Inhalte des **X**- und **Y**-Registers. Es sind:

- 9** **X≠Y** Sind die Inhalte von **X** und **Y** ungleich?
- 9** **X≤Y** Ist der Inhalt von **X** kleiner als oder gleich dem Inhalt des **Y**-Registers?
- 9** **X=Y** Sind die Inhalte von **X** und **Y** gleich?
- 9** **X>Y** Ist der Inhalt des **X**-Registers größer als der des **Y**-Registers?

In all diesen Fällen fährt der Rechner mit der sequentiellen Ausführung der nachfolgenden Programmschritte fort, falls die Antwort auf die entsprechende Frage *JA* ist, d. h. die Bedingung erfüllt ist. Ist dagegen die Antwort auf die entsprechende Frage *NEIN*, also die Bedingung nicht erfüllt, überspringt der Programmschritt-Anzeiger die nächsten beiden Schritte bevor er in der Programmausführung fortfährt.

In diesem Programmsegment wird bei Erreichen von **9** **X>Y** der Inhalt von **X** mit dem von **Y** verglichen.

1. Der Inhalt des **X**-Registers ist *größer* als der des **Y**-Registers. Dann wird **GTO** **9** ausgeführt und der voranstehende Abschnitt wiederholt.
2. Der Inhalt des **X**-Registers ist *nicht größer* als der des **Y**-Registers. Dann wird **GTO** **9** übersprungen und **+** **3** ausgeführt.

Immer wenn im Rahmen eines Programms eine Testoperation ausgeführt wird, speichert der Rechner den Wert in **X** außerdem im Register **R₉** ab. Wenn Sie **R₉** als Speicherregister verwenden wollen und das Programm Vergleichsoperationen enthält, müssen Sie diesen Punkt besonders beachten.

In der Folge wollen wir einige Programme schreiben, die solche numerische Vergleichsoperationen beinhalten.

ZWEI PROGRAMME MIT NUMERISCHEN VERGLEICHSD- OPERATIONEN


Das erste Programm bezieht sich auf die folgende Anekdote:

«Es gab einmal vor vielen Jahren ein Königreich, das von einem müden gelangweilten König regiert wurde. Immer auf der Suche nach neuen Vergnügen, ließ er eines Tages die folgende Nachricht in seinem Königreich verbreiten: <Derjenige, der dem König ein Spiel beibringt, mit dem er sich wahrhaft vergnügen kann, soll nach seiner Wahl einen Wunsch frei haben!>. Es meldete sich bald darauf ein junger Mann, der dem König das Schachspiel zeigte. Der war davon so begeistert, daß er den Mann sogleich frug: <So sagt denn, was ist Euer Wunsch?>. Der junge Mann antwortete: <Oh, weiser und edler König, alles was ich wünsche ist, daß Ihr ein einziges Weizenkorn auf das erste Feld des Schachbretts legt, auf das zweite Feld die doppelte Menge, auf das dritte doppelt soviel wie auf das zweite, und so weiter bis zum letzten Feld. Gebt mir dann die Weizenkörner, die Ihr auf das letzte der Felder des Schachbretts legt!>.

<Dies ist ein wohlhabendes Königreich>, antwortete der König verwundert. <Sicher habt Ihr einen weiteren Wunsch?>. – Aber der junge Mann blieb bei seiner einzigen Bitte.»

P.S. – Die Jahresproduktion des Königreichs an Weizen betrug 1 000 000 000 Körner (1 Milliarde), und das Schachbrett hat 64 Felder, die zu füllen waren.

War der junge Mann wirklich so bescheiden, wie es den Anschein hat?

Das Programm berechnet nacheinander die für jedes Feld benötigte Anzahl Weizenkörner und speichert sie jeweils in R_2 . In R_1 ist die jeweilige Nummer des Schachfeldes abgespeichert. Sollten auf eins der Felder mehr als eine Milliarde Körner gelegt werden müssen, hält das Programm an und zeigt die Nummer dieses Feldes an. Können dagegen alle 64 Felder gefüllt werden, ohne daß dem Königreich dadurch seine Existenz genommen wird, soll das Programm anhalten und die Zahl der Weizenkörner anzeigen, die zu bezahlen sind. Schalten Sie in den W/PRGM-Modus, drücken Sie  PRGM zum Löschen des

Programmspeichers und geben Sie die nachstehende Tastenfolge ein.

Tasten	Bemerkungen
LBL A 1 STO 1 STO 2 LBL 1 2 STO x 2 RCL 1 1 + STO 1 6 4 g x=y RCL 2 RTN RCL 2 EEX 9 g x>y GTO 1 RCL 1 RTN	<p>Programmbeginn</p> <p>Vorbesetzen von R_1 und R_2, um erstes Feld zu berücksichtigen</p> <p>Beginn der Wiederholung</p> <p>Berechnet die jeweilige Anzahl</p> <p>Erhöht die Nummer des Feldes um eins</p> <p>Bereits Feld Nr. 64? Falls ja, zeige Anzahl Körner an und halte an</p> <p>Anderenfalls vergleiche Anzahl mit 1 Milliarde (10^9)</p> <p>Sprung, falls eine Milliarde größer als die Anzahl</p> <p>Anderenfalls halte an und zeige die Nummer des Feldes an</p>

Schalten Sie zurück in den RUN-Modus und führen Sie Ihr Programm aus. Nach wenigen Sekunden sollten Sie die folgende Anzeige erhalten: **31.00**.

Mehr als eine Milliarde Körner mußten bereits auf das 31. Feld gelegt werden! (Wollen Sie die genaue Anzahl wissen, drücken Sie **RCL** **2**.)

Um die Anzahl Weizenkörner zu berechnen, die auf das 64. Feld gelegt werden müßte, drücken Sie **2** **ENTER** **64** **9** **y^x**. Es erübrigt sich wohl zu erwähnen, daß der besagte junge Mann hingerichtet wurde!

Das zweite Programm berechnet den Arkussinus (arcsin) eines Ausgangswertes x (x muß innerhalb der Grenzen -1 und $+1$ liegen). Das Programm prüft den so erhaltenen Winkel und macht ihn positiv, indem es 360 Grad addiert, sofern der Winkel kleiner oder gleich Null ist.

Schalten Sie wieder in Stellung W/PRGM, drücken Sie **f** **PRGM** und tasten Sie folgendes Programm ein.

Tasten	Bemerkungen
LBL }	Programmbeginn
B }	
f1 }	Berechnet arcsin x
SIN }	
0	Schreibt 0 in X
9 x^zy	Tauscht 0 und arcsin x aus
9 x[≤]y	Prüft den Winkel
3 }	Diese beiden Schritte werden
6 }	übersprungen, wenn der Winkel
0 }	positiv ist
+ }	Addiert 0 oder 360, um sicher-
RTN	zustellen, daß der Winkel positiv ist

Wenn der Winkel positiv ist, wird 3 und 6 übersprungen und nur 0 zu dem Wert addiert, was keine Auswirkung hat. Anderenfalls wird 360 addiert. Wir wollen eine Aufgabe rechnen.

Beispiel: Berechnen Sie den Arkussinus von 0.5 und -0.5 .

Drücken Sie

.5

B

.5 **CHS**

B

Anzeige

.5

30.00 (Grad)

-.5

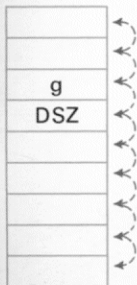
330.00

DEKREMENT UND SPRUNG BEI NULL

Der Tastenbefehl **DSZ** (Decrement and Skip on Zero) subtrahiert 1 vom Inhalt des Registers R_8 und testet dann auf *ungleich Null*. Die Abfrage kann wie folgt formuliert werden:

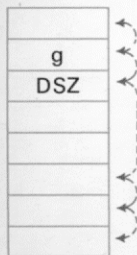
«Ist der Inhalt von Register R_8 eine andere Zahl als Null?».

Auch hier überspringt das Programm die nachfolgenden zwei Schritte, wenn die Bedingung nicht erfüllt ist, d.h. R_8 nach Abzug von 1 gleich Null ist. Anderenfalls läuft die Programmausführung sequentiell weiter.



Bedingung erfüllt, falls Wert in $R_8 \neq 0$.

Programmausführung läuft Schritt für Schritt weiter.



Bedingung nicht erfüllt, falls Inhalt von $R_8 = 0$.

Programmschritt-Anzeiger überspringt zwei Speicherpositionen, bevor die Ausführung des Programms weitergeht.

Wenn Sie innerhalb eines Programmes **DSZ** verwenden, ist es normalerweise nicht sinnvoll, das Register R_8 für andere Speicherzwecke zu verwenden. Die **DSZ**-Operation erfordert im übrigen, daß der Inhalt von R_8 innerhalb der Grenzen $-10^{10} \leq r_8 \leq 10^{10}$ liegt.

DSZ kann in Ihren Programmen auf verschiedene Weise eingesetzt werden. Es kann als Zähler, als Flag (siehe Seite 91) oder dazu verwendet werden, einen Teil oder das gesamte Programm zu wiederholen.

EIN PROGRAMM UNTER VERWENDUNG VON **DSZ**

Wenn Sie **DSZ** in Ihrem Programm als Zähler verwenden wollen, speichern Sie Null in Register R_8 und fügen Sie den **DSZ**-Befehl in den Programmteil ein, der wiederholt werden soll. Während der Ausführung des Programms enthält R_8 dann die Nummer des Schleifendurchlaufs (allerdings als negative Zahl).

Das folgende Programm berechnet die Summe und den Mittelwert einer Gruppe von Daten unter Verwendung von **DSZ**. Schalten Sie in Stellung W/PRGM, löschen Sie den Programmspeicher mit **f** **PRGM** und tasten Sie die Programmschritte ein.

Tasten	Bemerkungen
LBL	Start des Vorbereitungsprogramms
A	
f	
REG	Löscht alle Register
RTN	
LBL	
B	Programmbeginn für Summation der Daten
STO	
+	
1	Jeder Wert wird zum Inhalt von R_1 addiert
g	
DSZ	
	Der Inhalt von R_8 wird jeweils um 1 verringert

90 Programmierung

RCL	1	}
RTN		
LBL		
C		
RCL	1	}
RCL	8	
CHS		
RTN		

Anzeige der laufenden Summe

Programmbeginn für Bildung
des Mittelwertes

Gesamtsumme wird durch die
positive Anzahl der Schleifen-
durchläufe dividiert

Das erste, der Taste **A** zugeordnete Programm löscht lediglich alle Register. Das Programm **B** addiert jede Zahl zu den vorangegangenen und zeigt die jeweilige laufende Summe an.

DSZ wird verwendet, um die jeweilige Nummer des Schleifendurchlaufs zu zählen. In R_8 steht als negative Zahl die Anzahl der inzwischen aufaddierten Werte. Da bei **DSZ** die Prüfung der Sprungbedingung jeweils erst nach Subtraktion der 1 von dem Inhalt in R_8 erfolgt und der Registerinhalt r_8 mit Null vorbesetzt wurde, ist die Bedingung ($r_8 \neq 0$) immer erfüllt, und das Programm fährt sequentiell fort. Das dritte, unter **C** abgespeicherte Programm, errechnet den Mittelwert, indem es die Gesamtsumme durch die Anzahl der Schleifendurchläufe dividiert. Da der Wert in R_8 negativ ist, wird erst das Vorzeichen gewechselt, bevor die Division ausgeführt wird.

Schalten Sie jetzt wieder zurück in Stellung RUN und rechnen Sie die folgende Aufgabe.

Beispiel: Berechnen Sie die Summe und den Mittelwert der folgenden Zahlenreihe: **65 78 908 345 23 98**

Drücken Sie

Anzeige

A	0.00	Vorbereitungsprogramm
65 B	65.00	Laufende Summe
78 B	143.00	Laufende Summe
908 B	1051.00	Laufende Summe
345 B	1396.00	Laufende Summe
23 B	1419.00	Laufende Summe
98 B	1517.00	Endsumme
C	252.83	Mittelwert

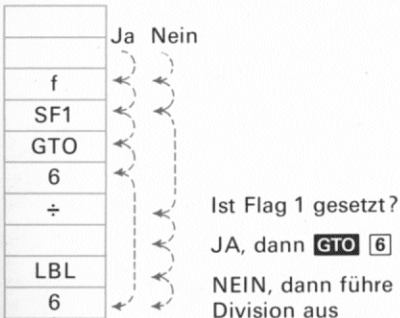
FLAGS

Ihr HP-65 verfügt außerdem über zwei «Flags», unsichtbare Einrichtungen im Innern des Rechners, die zwei verschiedene Zustände annehmen können; Sie können die Flags «setzen» oder «löschen». Eine andere Tastenfolge prüft die Flags und stellt fest, ob sie gesetzt sind oder nicht. Die Eigenschaften der Flags kann man mit denen eines Schalters vergleichen, der ja auch nur die beiden Zustände «ein-» oder «ausgeschaltet» kennt. In der nachstehenden Aufstellung sind die Tasten zusammengestellt, die die Flags setzen, löschen und prüfen:

Tasten	Funktion
f SF1	Setzt Flag 1
f1 SF1	Löscht Flag 1
f TF1	Prüft Flag 1. Ist Flag 1 gesetzt?
f1 TF1	Prüft Flag 1. Ist Flag 1 gelöscht?
f SF2	Setzt Flag 2
f1 SF2	Löscht Flag 2
f TF2	Prüft Flag 2. Ist Flag 2 gesetzt?
f1 TF2	Prüft Flag 2. Ist Flag 2 gelöscht?

Diese vier Anweisungen können von einem Programm oder vom Tastenfeld her ausgeführt werden

Wie bei den bisherigen Vergleichs-Operationen läuft auch hier die Programmausführung sequentiell weiter, wenn die Antwort auf die jeweilige Testfrage *JA* ist. Ist die Bedingung dagegen nicht erfüllt, die Antwort auf die Testfrage also *NEIN*, überspringt der Programmschritt-Anzeiger die beiden nächsten Speicherpositionen bevor die Programmausführung weiterläuft.



Die Flags verbleiben so lange in ihrem Zustand, bis eine Operation (vom Tastenfeld oder vom Programm) ausgeführt wird, die den Zustand ändert.

EIN PROGRAMM UNTER VERWENDUNG DER FLAGS

Die Beziehungen zwischen Geschwindigkeit, Zeit und zurückgelegter Wegstrecke für einen bewegten Körper werden durch die folgenden Gleichungen ausgedrückt:

$d = s \cdot t$ Wegstrecke = Geschwindigkeit \times Zeit
 $s = d / t$ Geschwindigkeit = Weg / Zeit
 $t = d / s$ Zeit = Wegstrecke / Geschwindigkeit

Wir wollen ein Programm schreiben, das eine beliebige dieser Größen berechnet, soweit die beiden anderen gegeben sind.

Schalten Sie in Stellung W/PRGM, drücken Sie **f** **PRGM** und geben Sie die nachstehende Tastenfolge ein.

Tasten	Bemerkungen
<div><div>LBL</div><div>A</div><div>f</div><div>SF1</div><div>f</div><div>STK</div></div>	<div>Start der Vorbereitungsroutine</div> <div>Setzt Flag 1</div> <div>Löscht den Stack</div>

DSP

•

4

RTN

LBL

B

f

TF1

STO 1

RTN

RCL 2

RCL 3

x

RTN

LBL

C

f

TF1

STO 2

RTN

RCL 1

RCL 3

÷

RTN

LBL

D

f

TF1

GTO

1

RCL 1

RCL 2

÷

f

→DMS

RTN

LBL

1

Ändert das Anzeigeformat

Beginn der Entfernungsroutine

Prüft, ob Flag 1 gesetzt ist

Falls ja, speichere d und halte an

Sonst berechne d

Beginn der Geschwindigkeits-
routine

Prüft, ob Flag 1 gesetzt ist

Falls ja, speichere Geschwin-
digkeit und halte an

Sonst berechne s

Beginn der Zeitroutine

Prüft, ob Flag 1 gesetzt ist

Falls ja, Sprung nach Marke 1

Sonst berechne t

Wandelt t in Stunden, Minuten
und Sekunden um

<div>f1</div> <div>→DMS</div> <div>STO 3</div> <div>RTN</div> <div>LBL</div> <div>E</div> <div>f1</div> <div>SF1</div> <div>RTN</div>	<p>Wandelt eingegebenes t in dezimale Stunden um Speichert t in R_3</p> <p>Beginn der Rechenroutine</p> <p>Löscht Flag 1</p>
---	---

Das Programm zur Berechnung der Zeit ist so aufgebaut, daß der Wert in Stunden, Minuten und Sekunden eingegeben wird.

Diesen Wert wandelt das Programm für die Berechnungen in dezimale Stunden um. Um eine der Variablen zu berechnen, drücken Sie als erstes **A**, worauf – als Vorbereitungsschritt – Flag 1 gesetzt wird. Geben Sie dann einen der bekannten Werte ein und drücken Sie die zugehörige Programmkontrolltaste. Da das Flag gesetzt ist, wird dieser Wert abgespeichert und nicht berechnet. Verfahren Sie ebenso mit dem zweiten bekannten Wert. Auch der wird lediglich gespeichert. Um nun den unbekannten Wert zu berechnen, drücken Sie **E** zum Löschen des Flag 1 und dann die zugehörige Programmkontrolltaste. Da jetzt das Flag gelöscht ist, wird dieser Wert nicht gespeichert, sondern berechnet. Nach jeder solchen Berechnung müssen Sie wieder **A** drücken, damit das Flag 1 erneut gesetzt wird.

Rechnen Sie einmal das nachstehende Beispiel durch, dann wird Ihnen die Funktionsweise deutlich.

Beispiel: Berechnen Sie s für $t = 5$ Stunden und 30 Minuten und $d = 500$ (beliebige Längeneinheiten).

Drücken Sie

A

5.30 **D**

500 **B**

E C

Anzeige

0.0000 Startet die Vorbereitungsroutine

5.5000 Zeit wird in dezimale Stunden
umgewandelt

500.0000 Entfernung (Strecke)

90.9091 Geschwindigkeit in Einheiten
pro Stunde

Beispiel: Berechnen Sie t für $s = 700$ und $d = 5000$.

Drücken Sie

A

700 **C**

5000 **B**

E D

Anzeige

0.0000 Startet die Vorbereitungsroutine

700.0000 Geschwindigkeit

5000.0000 Entfernung

7.0834 Zeit = 7 Stunden, 8 Minuten und 34 Sekunden

Beispiel: Berechnen Sie d für $s = 70$ und $t = 74$ Stunden, 42 Minuten und 50 Sekunden.

Drücken Sie

A

60 **C**

74.4250 **D**

E B

Anzeige

0.0000 Startet die Vorbereitungsroutine

70.0000 Geschwindigkeit

74.7139 Zeit in dezimalen Stunden

5229.9722 Ergebnis

Durch Drücken von **DSP** **•** **2** können Sie jetzt für die nachfolgenden Berechnungen wieder das Standard-Anzeigeformat einstellen (Festkommadarstellung mit 2 Nachkommastellen). Obwohl die Flags doch einigen Speicherraum für die Setz-, Löscho- und Abfrageoperationen belegen, sind sie doch sehr zweckmäßig zur Programmierung solcher Entscheidungen, die nicht unmittelbar auf dem Vergleichen von Zahlenwerten beruhen.

DSZ ALS FLAG

Sie können auf einfache Weise ein Flag konstruieren, das sich sogar selbst löscht, indem Sie Register R_8 mit 1 besetzen und den **DSZ**-Befehl verwenden. Wenn das Programm **DSZ** ausführt, wird der Inhalt von R_8 um 1 verringert und damit Null. Jetzt wird R_8 auf «ungleich Null» getestet. Da diese Bedingung nicht erfüllt ist, werden die beiden nächsten Schritte übersprungen. Die gleiche Wirkung hätte der Test, ob ein bestimmtes Flag gesetzt ist. Wird der **DSZ**-Befehl zum zweitenmal ausgeführt, führt der Programmschritt-Anzeiger (wie auch in allen weiteren Fällen) das Programm sequentiell aus (so, als ob ein nicht gesetztes Flag getestet würde), da jetzt der Inhalt von R_8 ungleich Null ist.

PROGRAMMUNTERBRECHUNGEN

R/S (Run/Stop) ist ein spezieller Tastenbefehl, der im Rahmen eines Programms anders wirkt als vom Tastenfeld aus gedrückt. Als Bestandteil des Programms hält **R/S** den Programmablauf an. Damit ist die Eingabe von Daten möglich, es können Zwischenrechnungen durchgeführt werden usw. Vom Tastenfeld aus startet **R/S** die Programmausführung von der Stelle aus, an der der aktive Programmschritt-Anzeiger gerade steht. **R/S** kann auch verwendet werden, um Programme auf andere Weise zu kontrollieren, als das bisher geschehen ist. Die nachfolgenden Informationen sollten Sie daher genau beachten.

EINGEBEN VON DATEN

Der Befehl **R/S** wird hauptsächlich dazu benutzt, das Programm anzuhalten, damit Daten eingegeben werden können. Wenn innerhalb eines laufenden Programms ein **R/S** auftritt, hält das Programm an, wobei der Programmschritt-Anzeiger an der Stelle des **R/S** stehenbleibt. Die Programmausführung kann jetzt jederzeit wieder gestartet werden, indem vom Tastenfeld aus **R/S** gedrückt wird.

DIE VERWENDUNG VON **R/S** ZUM EINTASTEN VON DATEN

Um Ihnen die Wirkungsweise zu veranschaulichen, wollen wir ein Programm schreiben, das die Gesamtkosten verschiedener Artikel mit unterschiedlichem Preis nach Abzug von 15% Rabatt berechnet.

Schalten Sie um in Stellung W/PRGM und drücken Sie **f** **PRGM**. Geben Sie dann die folgende Programmliste ein:

Tasten	Bemerkungen
LBL	} Programmbeginn
A	
f	
STK	
	Vorbereitungsschritt

LBL	}	Markierung der Stelle, ab der wiederholt werden soll
3	}	Halt zur Eingabe der Menge
R/S		Kopiert Menge nach Y
ENTER↑		Halt zur Eingabe des Preises
R/S		Menge × Preis
×		
•	}	Berechnung des Nettopreises
8	}	
5	}	
×		Addition zur Gesamtsumme
+		
GTO	}	Wiederholung ab Marke 3
3	}	

Beachten Sie insbesondere, daß am Ende dieses Programms kein **RTN** steht. Da das Programm ohnehin eine niemals endende Schleife bildet, ist das hier nicht nötig. Außerdem hält das Programm sowieso bei jedem Schleifendurchlauf an, damit Sie neue Daten eingeben können.

Wenn das Programm zum erstenmal anhält, tasten Sie die Menge des jeweiligen Artikels ein und starten das Programm wieder mit **R/S**. Es läuft dabei immer ab der Stelle weiter, wo der aktivierte Programmschritt-Anzeiger gerade steht. Beim nächsten Halt geben Sie den Einzelpreis des Artikels ein und starten erneut mit **R/S**. Das Programm errechnet daraufhin den Einzelpreis, addiert ihn zu den Gesamtkosten, springt zurück zur Marke 3 und hält dann zur Eingabe neuer Daten an. Dabei wird die laufende Summe angezeigt.

Schalten Sie in Stellung RUN und führen Sie das Programm aus.

Beispiel: Nehmen Sie an, daß Sie auf den folgenden Einkäufen 15% Rabatt erhalten:

Menge	Einzelpreis
5	DM 2,—
7	DM 4,—
8	DM 5,—
22	DM 6,—

98 Programmierung

Berechnen Sie die Gesamtkosten.

Drücken Sie	Anzeige	
A	0.00	Der Stack ist gelöscht
5 R/S	5.00	Die erste Mengenangabe
2 R/S	8.50	Laufende Summe
7 R/S	7.00	
4 R/S	32.30	Laufende Summe
8 R/S	8.00	
5 R/S	66.30	Laufende Summe
22 R/S	22.00	
6 R/S	178.50	Gesamtkosten

Geht dem **R/S** innerhalb eines Programms unmittelbar die Eingabe einer Zahl in das **X**-Register voraus, so wird diese Zahl bei dem anschließenden Eintasten von Daten überschrieben und nicht, wie in allen anderen Fällen, im Stack angehoben. Damit ist es möglich, eine Zahl in das Anzeige-(**X**-)Register zu schreiben, die angibt, welche Daten bei dem nun folgenden Programmstopp einzutasten sind, ohne daß diese Zahl in die folgenden Berechnungen einbezogen wird. Im übrigen beeinflusst **R/S** den Stack-Lift nicht.

Anmerkung: Zifferntasten, die im Programm unmittelbar auf ein **R/S** folgen, sollten von diesem Schritt durch ein **ENTER** getrennt werden*.

PROGRAMMKONTROLLE MITTELS **R/S**

Bis hierher hat jedes der Programme, die Sie geschrieben haben, mit einem **LBL** begonnen und mit **RTN** geendet. Wir haben diesen Weg als ersten gezeigt, weil er in den meisten Fällen der bequemste ist und häufig angewendet wird. Die Vielseitigkeit Ihres HP-65 läßt aber auch andere Möglichkeiten zu. **R/S** kann sehr effektiv zur Steuerung von Vorbereitungsroutinen und selbst ganzer Programme verwendet werden, ohne daß eine einzige Marke oder ein **RTN** benötigt werden. Häufig läßt sich damit auch der Speicherbedarf eines Programms ein-

* Sie sollten für die fortgeschrittene Programmierung wissen, wie **R/S**, **RTN** und die Programmschritt-Anzeiger im Innern des Rechners wirken. Dieses Gebiet wird in Anhang B behandelt.

schränken, und man spart sich u.U. den Einsatz der Programmkontrolltasten für andere Stellen größerer Programme auf.

Die Regel für die Verwendung von **R/S** ist einfach: Paaren Sie jeweils ein **R/S** mit einem **R/S**. Wenn Sie also das Programm mit **R/S** starten wollen, müssen Sie einen zweiten **R/S**-Befehl im Programm einbauen, der das Programm wieder anhält. Genauso mußten Sie ein Programm, das mit **LBL** (**A** bis **E**) beginnt und mit **RTN** endet, durch Drücken der entsprechenden Programmkontrolltaste starten.

Programme, die mit **R/S** kontrolliert werden, sollten am Speicheranfang stehen, damit sie mit der Tastenfolge **RTN R/S** leicht aufgerufen werden können.

Anmerkung: Programme, die mit einer Marke beginnen, sollten nicht mit **R/S** gestartet werden. Der Grund dafür wird auf Seite 108 unter *Schrittweises Vorgehen mittels SST* behandelt.

EIN MITTELS **R/S** KONTROLLIERTES PROGRAMM

Wir wollen jetzt ein Programm schreiben, das mit **R/S** gesteuert wird. Schalten Sie in Stellung W/PRGM und drücken Sie **f** **PRGM**. Tasten Sie dann das folgende Programm ein, das den Ausdruck $\sqrt{x^2 + y^2}$ berechnet.

Tasten	Bemerkungen
ENTER ↑	Speichert gerade eingegebenes x
R/S	Halt zum Eintasten von y
f1 } √x }	Berechnet y^2
g x²y	
f1 } √x }	Berechnet x^2
+	
f	
√x	
R/S	Programmstopp

100 Programmierung

Beachten Sie insbesondere hier das Fehlen einer Marke am Beginn und des **RTN** am Ende des Programms.

Schalten Sie zur Ausführung des Programms zurück in Stellung RUN. Drücken Sie jetzt **RTN**, um den Programmschritt-Anzeiger an den Speicheranfang zu setzen, geben Sie einen Wert für x ein und drücken Sie **R/S**. Wenn das Programm anhält, geben Sie den Wert für y ein und drücken Sie erneut **R/S**. Das Programm hält dann wieder zur Anzeige des Resultates an.

Testen Sie das Programm anhand des nachstehenden Beispiels.

Beispiel: Berechnen Sie $\sqrt{7^2 + 9^2}$.

Drücken Sie	Anzeige
RTN	0.00
7 R/S	7.00
9 R/S	11.40 Ergebnis

VERSCHIEDENE ARTEN DES PROGRAMM-STOPPS

Anhalten mit blinkender Anzeige. Treten innerhalb eines Programms Fehler auf, die zur Anzeige blinkender Nullen führen, so wird das Programm gleichzeitig angehalten. Zum Aufheben der Fehleranzeige (Blinken) können Sie eine beliebige Taste drücken (zweckmäßigerweise **CLX**). Um den Grund für den Fehler zu finden, schalten Sie kurzfristig in den W/PRGM-Modus. Sie sehen dann die zuletzt ausgeführte Operation in Form ihres Programmcodes. Außerdem können Sie noch über **g** **LST X** den letzten X -Wert zurückrufen.

Normales Anhalten. Das normale Anhalten eines Programms, d.h. verursacht durch **RTN** oder **R/S**, können Sie daran erkennen, daß in Stellung W/PRGM die Codes 24 oder 84 angezeigt werden.

Versehentliches Anhalten. Beachten Sie, daß das Drücken einer beliebigen Taste ein laufendes Programm anhält. Achten Sie bei der Ausführung eines Programms darauf, daß dies nicht unbeabsichtigt geschieht.

Markierter Programmstopp. Sofern es der Speicherplatz erlaubt, ist es oft sinnvoll, eine bekannte Zahl in das **X**-Register zu schreiben, bevor das Programm zur Entgegennahme von Daten anhält. Auf diese Weise können die Daten angegeben werden, die einzutasten sind. Wenn Ihr Programm zum Beispiel acht Ausgangswerte nacheinander annimmt, ist es nützlich, nacheinander die Zahlen 1 bis 8 als Signal für den gerade benötigten Wert in der Anzeige zu erhalten.

Wenn das Programm solche Kennzahlen unmittelbar vor Ausführung eines **R/S** in das **X**-Register schreibt, werden sie bei der nachfolgenden Dateneingabe überschrieben. In allen anderen Fällen (Abruf aus einem Register, Ergebnis einer Rechnung) werden diese Inhalte des **X**-Registers beim Eintasten von Daten im Stack angehoben.

Anhalten wegen Speicherüberlauf. Wenn im Rahmen eines laufenden Programms eine Rechnung zu einem Ergebnis führt, das den Zahlenbereich des Rechners übersteigt, hält das Programm an. Als Anzeige erhalten Sie 9.999999999 99.

Anhalten wegen Speicherunterlauf. Tritt im Laufe von Berechnungen ein Ergebnis auf, das zu klein ist ($<10^{-99}$), um in einem Register gespeichert werden zu können, hält ein laufendes Programm an. In diesem Fall erhalten Sie als Anzeige 0.00. (Anzahl der Nachkomma-Nullen entsprechend dem gewählten Anzeigeformat.)

PROGRAMME ZUR LÖSUNG IHRER RECHENPROBLEME

Sicherlich haben Sie beim Lesen dieses Handbuches einiges über die Programmierung Ihres HP-65 lernen können, zumal dann, wenn Sie die verschiedenen Beispiele selbst einmal durchgerechnet haben. Jetzt wollen Sie sich zweifellos an die Lösung Ihrer eigenen Aufgabenstellungen begeben. Dieser Abschnitt verfolgt den Zweck, Ihnen einen möglichen Weg aufzuzeigen, wie Sie sich der Lösung Ihrer Aufgaben nähern können.

Im allgemeinen besteht die Erstellung eines Programms aus drei verschiedenen Abschnitten:

102 Programmierung

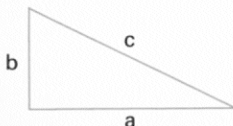
1. Definieren des Problems.
2. Entscheiden, wie das Problem gelöst werden soll.
3. Festlegen der Tastenfolge.

Über Punkt 3 haben Sie inzwischen einiges gelernt. Wenden wir uns also den Punkten 1 und 2 zu.

Beispiel für das Schreiben eines Programms:

1. Definition des Problems: Welche genaue Aufgabe soll das Programm lösen?

Schreiben Sie ein Programm für den «Satz des Pythagoras»:



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

2. Entscheiden Sie sich, wie die Aufgabe gelöst werden soll. Auf welche Weise würden Sie die Aufgabe auf dem Papier lösen? Sie müssen dafür bestimmen, welche Größen das Programm berechnen soll und welche ihm vorgegeben werden können. Sie müssen außerdem festlegen, welche Programmkontrolltasten Sie einsetzen möchten und wie Sie sie verwenden wollen.

Eine Möglichkeit, diese Fragen zu entscheiden, besteht darin, daß Sie sich die Beschriftung der Magnetkarte vorgeben. Wollen Sie im vorliegenden Beispiel nur «c» berechnen, könnte die Karte folgendermaßen aussehen:

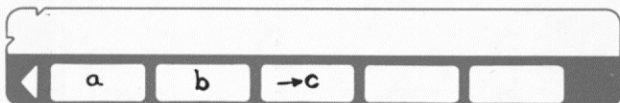


Wenn Sie Ihr Programm auf diese Weise schreiben, müssen Sie erst einen Wert für «a» eingeben und dann **ENTER** drücken. Dann tasten Sie «b» ein und drücken **A**. Natürlich können Sie die Aufgabe auch über eine Karte lösen, die folgendermaßen aussieht:



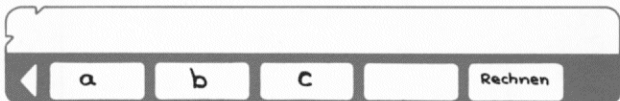
Sie müßten einen Wert für «a» eingeben und dann **A** drücken, anschließend «b» eintasten und dann **B** drücken. Auf diese Weise könnten Sie «a» speichern, so daß Sie den Wert für weitere Berechnungen mit verschiedenen «b»-Werten nicht erneut einzugeben brauchten.

Ein dritter Weg zur Lösung des Problems könnte auf der Magnetkarte wie folgt aussehen:



Hier könnten Sie sowohl «a» als auch «b» abspeichern und müßten nur den Wert eingeben, der sich gegenüber der vorhergehenden Rechnung geändert hat. Sie müßten einen Wert für «a» eintasten, **A** drücken, «b» eingeben und **B** drücken und könnten dann mit **C** die Berechnung der gesuchten Seite «c» auslösen.

Vielleicht möchten Sie die Gleichung aber auch für jede der drei Variablen lösen können, vorausgesetzt, die beiden anderen sind Ihnen gegeben. Für diesen Fall würde die Magnetkarte so beschriftet werden:



Sicher würde dieses Programm erfordern, daß Sie in einer Vorbereitungsroutine mit **RTN** **R/S** ein Flag setzen. Dann würden Sie nacheinander die beiden bekannten Variablen eintasten und jeweils die zugehörige Programmkontrolltaste drücken. Abschließend müßten Sie erst **E** und dann die Programmkontrolltaste des gesuchten Wertes drücken.

104 Programmierung

Sie erkennen daran, daß die Suche nach dem zweckmäßigen Lösungsweg ein kreativer Vorgang ist. Er hängt wesentlich davon ab, was Sie als Resultat wünschen und welche Daten zur Verfügung gestellt werden. Wie Sie sich dieser Lösung nähern, bestimmt im wesentlichen, wie nachher Ihr Programm aussieht.

- Schreiben Sie die einzelnen Tastenschritte auf, aus denen das Programm bestehen soll. Bei der ersten Niederschrift ist es oft sinnvoll, die Spalte «COMMENTS» (Bemerkungen) des Programmformulars dafür zu verwenden, die jeweiligen Inhalte der X-, Y-, Z- und T-Register zu verfolgen. In der endgültigen Fassung können Sie statt dessen Kommentare einfügen, die Ihnen später einmal die Aufgabe bestimmter Programmteile auf einen Blick erklärt.

Bei dem folgenden Programm sind wir nach der dritten Methode vorgegangen. Sowohl «a» als auch «b» werden gespeichert, bevor «c» berechnet wird.

Schalten Sie in Stellung W/PRGM, drücken Sie **f** PRGM und geben Sie die nachstehende Tastenfolge ein.

Tasten	Bemerkungen
LBL }	Beginn des Programms zum
A }	Speichern von «a»
STO 1	Speichere «a»
RTN	
LBL }	Beginn des Programms zum
B }	Speichern von «b»
STO 2	Speichere «b»
RTN	
LBL }	Beginn der Berechnung von «c»
C }	
RCL 1	Rufe «a» zurück
ENTER ↕ }	Berechne a^2
x }	
RCL 2	Rufe «b» zurück
ENTER ↕ }	Berechne b^2
x }	

<div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">+</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">f</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 2px;">\sqrt{x}</div> <div style="border: 1px solid black; padding: 2px;">RTN</div> </div>	$a^2 + b^2$ Resultat
--	-----------------------------

Schalten Sie zurück in Stellung RUN und prüfen Sie nach, ob das Programm einwandfrei arbeitet.

Beispiel: Berechnen Sie c für a = 10 und b = 5,
 für a = 78 und b = 22
 und für a = 78 und b = 10.

Drücken Sie	Anzeige
10 A	10.00 Wert für a
5 B	5.00 Wert für b
C	11.18 Resultat
78 A	78.00 Wert für a
22 B	22.00 Wert für b
C	81.04 Resultat
10 B	10.00 Geben Sie hier nur b ein
C	78.64 Resultat

Schalten Sie wieder um in den W/PRGM-Modus und zeichnen Sie das Programm auf eine ungeschützte Magnetkarte auf. Anschließend können Sie die Karte wie geplant beschriften.

ENTWICKELN EINES FLUSSDIAGRAMMS

Eine wertvolle Hilfe bei der Entscheidung, wie ein bestimmtes Problem gelöst werden soll, ist die Entwicklung eines sogenannten «Flußdiagramms». Als Zwischenschritt bei der Entwicklung eines Programms hilft Ihnen dieses Fluß- oder Ablaufdiagramm bei der Ermittlung des günstigsten Lösungsweges. In diesem Stadium ist es aufgrund der Übersichtlichkeit noch sehr einfach, Änderungen am Lösungsgang vorzunehmen oder logische Fehler zu erkennen. Wenn nach mehreren Versuchen (die auch ein erfahrener Programmierer meist benötigt) eine endgültige Fassung des Flußdiagrammes entstanden ist, ist es leicht, daraus die für die Lösung der Aufgabe erforderliche Tastenfolge zu entwickeln.

Wenngleich jedes solche Ablaufdiagramm, das Sie sich als Zwischenschritt notieren, nützlich ist, sollten doch gewisse Konventionen bei der Erstellung des Flußdiagramms berücksichtigt werden. So werden die Schritte oder Anweisungen, mit denen ein Programm beginnt oder endet, in ovale Kästchen gesetzt, die meisten anderen Anweisungen in Rechtecke. Alle Kästchen werden durch Striche in der Reihenfolge des Programmablaufs miteinander verbunden. Mit Pfeilen wird angezeigt, in welcher Richtung die Ausführung des Programms fortschreitet. Entscheidungspunkte, ab denen das Programm verzweigt, werden als rhombenförmige Kästchen gezeichnet, in die man die Verzweigungsbedingung hineinschreibt. Lautet die Antwort auf die entsprechende Frage JA (die Bedingung ist erfüllt), geht es mit der nächsten Anweisung weiter. Anderenfalls werden zuvor zwei Programmschritte übersprungen.

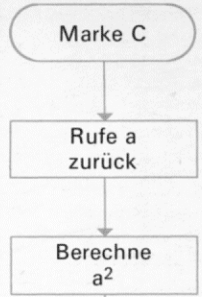
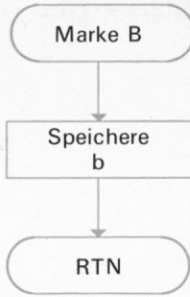
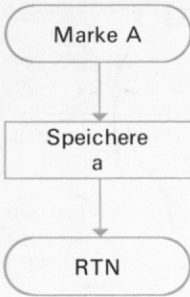
Auf der nächsten Seite finden Sie das Flußdiagramm zu unserem Programmbeispiel. Sie werden erkennen, daß es recht einfach ist, zu einem Flußdiagramm die entsprechende Tastenfolge zu finden.

Die ausführliche Behandlung des Themas «Flußdiagramme» würde hier zu weit führen. Wenn Sie sich für nähere Einzelheiten interessieren, dann nehmen Sie die entsprechende Fachliteratur zur Hilfe.

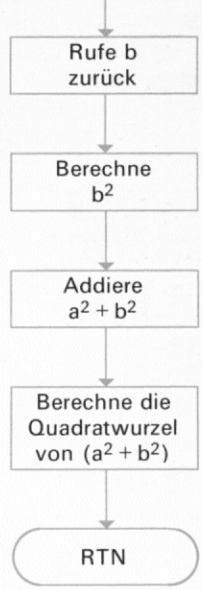
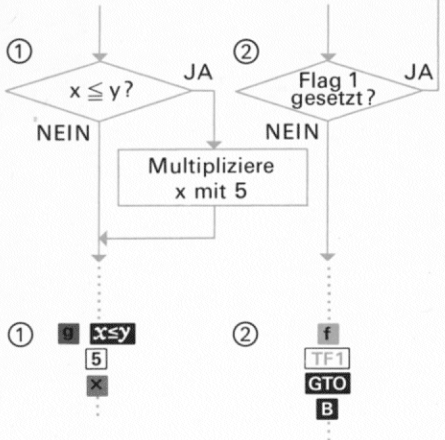
Jedes Programm, das Sie entwickeln, entsteht über die drei genannten Stufen. Mit einiger Übung werden Sie allerdings den Großteil dieser Arbeit im Kopf erledigen können.

ENDGÜLTIGE KORREKTUR IHRES PROGRAMMS

Auch wenn das eigentliche Programm bereits erstellt ist, findet selbst der erfahrenste Programmierer darin noch «schwache Stellen» oder gar Fehler. Einmal war das Flußdiagramm logisch falsch, ein anderes Mal sind bei der Eingabe des Programms Fehler unterlaufen. Jetzt gilt es, diese Fehler zu beseitigen. Die vielfältigen Korrekturmöglichkeiten, über die Ihr HP-65 verfügt, machen Ihnen das sehr leicht.



Beispiele für eine Programmverzweigung:



SCHRITTWEISES VORGEHEN MITTELS **SST**

Im RUN-Modus können Sie Ihr Programm durch Drücken von **SST** schrittweise ausführen und so den Funktionsablauf gewissermaßen in Zeitlupe studieren. Wenn offensichtlich nur bestimmte Teile Ihres Programms Fehler zu beinhalten scheinen, rücken Sie vorher den Programmschritt-Anzeiger auf die nächstliegende Marke.

Normalerweise wird auf Drücken der Taste **SST** hin nur ein einziger Programmschritt ausgeführt. Es gibt allerdings eine Ausnahme von dieser Regel. Erreichen Sie bei einem solchen schrittweisen Vortasten eine der Programmkontrolltasten im Programm, wird bei erneutem Drücken von **SST** der zweite Programmschritt-Anzeiger aktiviert, und das komplette Unterprogramm läuft ab, bevor der Rechner wieder anhält.

SST beendet nicht die Dateneingabe. Sie sollten daher einen **R/S**-Befehl im Programm durch **ENTER** von eventuell unmittelbar nachfolgenden Zifferntasten trennen. Anderenfalls kann es passieren, daß die vom Tastenfeld eingegebenen Zahlen mit denen zusammenlaufen, die im Programm nach dem **R/S** erzeugt werden. Dies gilt natürlich nur für das schrittweise Ausführen Ihres Programms mittels **SST**.

ZUSÄTZLICHE PROGRAMMSTOPPS ZUR FEHLER-ERKENNUNG

Wenn es der Speicherraum erlaubt, ist es oft von Nutzen, zusätzliche **R/S**-Befehle in das Programm einzubauen, um die Position des Programmschritt-Anzeigers bestimmen zu können. Dies gilt besonders für Unterprogramme, die sonst auf einen Druck von **SST** hin vollständig ablaufen, bevor die Kontrolle über das Programm an das Tastenfeld zurückgegeben wird. Nach Beseitigung aller Fehler können diese **R/S**-Befehle wieder aus dem Programm entfernt werden.

HÄUFIG VORKOMMENDE FEHLER

In der Folge sind die am häufigsten vorkommenden Fehler bei der Bedienung Ihres HP-65 zusammengefaßt.

PROGRAMMIERFEHLER

1. Es existieren mehrere identische Marken im Programm, da Sie versäumt haben, vor Eingabe des Programms **f** **PRGM** zu drücken.
2. Sie haben ein Programm versehentlich gelöscht, indem Sie in Stellung RUN eine Magnetkarte eingelesen haben.
3. Sie haben unabsichtlich das Programm auf einer Magnetkarte gelöscht, indem Sie die (ungeschützte) Karte in Stellung W/PRGM «eingelesen» haben.
4. Sie haben versehentlich unerwünschte Informationen in den Programmspeicher geschrieben, da der Modus-Wahlschalter in Stellung W/PRGM stand.
5. Der Speicher ist – nach Anzeige – schon bei weniger als 100 Schritten voll belegt, da Sie vergessen haben, zuvor die Ersatzprogramme zu löschen.
6. Sie haben nicht bedacht, daß eine Operation zu einem kombinierten Code zusammengefaßt wird und haben vergessen, ein **■** **NOP** einzufügen, falls diese beiden Schritte gegebenenfalls übersprungen werden sollen.
7. Sie haben irrtümlicherweise versucht, mit **LBL** (**0** bis **9**) Unterprogramme zu markieren. Dies geht nur in Verbindung mit den Programmkontrolltasten **A** bis **E**.
8. Sie haben versäumt, Flags vor deren Benutzung zu löschen.
9. Sie haben nicht bedacht, daß ein über Tastenfeld oder Programm aufgerufenes Unterprogramm die Inhalte von **LST X**, des Stack und der Speicherregister ändern kann.
10. Sie haben bei Verwendung von **DSZ** das Register R₉ falsch vorbesetzt.
11. Sie haben vergessen, daß das Register R₉ keinen kombinierten Code bildet oder daß sein Inhalt verlorengeht, wenn Sie trigonometrische Funktionen, Rechtwinklig/Polar-Umwandlungen oder numerische Vergleiche ausführen.
12. Programm- und Dateninformationen sind verlorengegangen, da Sie den Rechner versehentlich abgeschaltet oder das Netz-Ladegerät entfernt haben.

110 Programmierung

13. Sie haben versucht, von einem Unterprogramm aus eine zweite Routine aufzurufen.
14. Sie haben versäumt, beide Schritte eines nicht kombinierten Codes zu löschen.

RECHENFEHLER

1. Sie haben vergessen, die zum Aufruf einer bestimmten Funktion nötige Präfix-Taste (**f**, **f1** oder **g**) zu drücken.
2. Sie haben den Inhalt des **T**-Registers verloren, da Sie eine neue Zahl eingetastet oder aus einem Register zurückgerufen haben, womit der Stack angehoben wurde.
3. Während der Ausführung einer trigonometrischen Funktion war der falsche Winkel-Modus gesetzt.
4. Sie haben eine Operation ausgeführt, die sich auf das **X**- und **Y**-Register bezieht, und vergessen, deren Inhalte vorher mit **g** **x²y** zu tauschen.

ANHANG A

BETRIEBSGRENZEN

GENAUIGKEIT

Die Genauigkeit des HP-65 hängt von der jeweiligen Rechenoperation ab; bei den transzendenten Funktionen hängt sie außerdem vom Argument ab und ist nicht so ohne weiteres anzugeben. Daher sind die Genauigkeitsgrenzen keine exakten Angaben über die zu erwartenden Fehler, sondern sollen vielmehr als Anhalt für die Eigenschaften des Rechners dienen. Die hier genannten Genauigkeitsgrenzen geben, nach unserem besten Wissen, die maximalen Fehler für die entsprechenden Funktionen an.

Die elementaren Operationen \square $\boxed{1/x}$, \square $\boxed{\sqrt{x}}$, \square $\boxed{\rightarrow DMS}$ sind mit einem maximalen Fehler von ± 1 an der zehnten Stelle behaftet. Es ist durch die Rundung der Resultate auf 10 Stellen bedingt.

Einen solchen Rundungsfehler erhalten Sie beispielsweise bei der Berechnung von $(\sqrt{5})^2$. Wenn Sie $\sqrt{5}$ auf 10 wesentliche Stellen runden, erhalten Sie 2.236067977. Das Quadrieren dieser Zahl liefert ein 19stelliges Produkt 4.999999997764872529. Die entsprechende Rundung auf 10 Stellen ergibt 4.999999998. Wenn Sie jetzt die nächstgrößere Näherung (2.236067978) quadrieren, erhalten Sie 5.000000002237008484 oder, auf 10 Stellen gerundet, 5.000000002. Es gibt eben keine Zahl, deren Quadrat auf 10 Stellen gerundet genau gleich 5.000000000 ist.

Wenn Sie Zahlen mit zehn wesentlichen Stellen subtrahieren, dann beträgt der Fehler im Resultat ± 1 an der 10. Stelle des algebraisch größeren Operanden.

Die Fakultäts-Funktion ist bis auf ± 1 in der neunten Stelle genau. Werte, die in Grad-Minuten-Sekunden umgewandelt

112 Betriebsgrenzen

wurden, sind auf ± 1 Sekunde genau, ebenso die Ergebnisse von **f** **DMS+** und **f⁻¹** **DMS+**.

Die Genauigkeit der verbleibenden Operationen (trigonometrische, logarithmische und Exponentialfunktion) hängt vom jeweiligen Argument ab. Für ein Ausgangsargument, dessen Wert innerhalb von $\pm N$ in der 10. Stelle (siehe untenstehende Tabelle) des tatsächlichen Ausgangswertes ist, ist die Anzeige exakt. So liefert beispielsweise die Berechnung von $\ln(5)$ auf dem HP-65 den Wert 1.609437912. Dies ist allerdings eine Näherung, da dieser angezeigte Wert der Logarithmus einer Zahl ist, die zwischen 4.999999998 und 5.000000002 liegt, d.h. eine Abweichung von ± 2 ($N=2$ für Logarithmus) für die Ausgangsgröße in der zehnten Stelle.

Operation

Werte für N

f **LOG** }
f **LN** }
f⁻¹ **LN** }

2*

Trigonometrische Funktionen

3**

g **y^x**

4 für y, 10 für x

f⁻¹ **LOG**

7

f **R \rightarrow P** }

4

f⁻¹ **R \rightarrow P** }

* Das *angezeigte Resultat* ist in der 10. Stelle zusätzlich mit einer Ungenauigkeit von ± 1 für **f** **LN** und ± 3 für **f** **LOG** behaftet.

** Für die trigonometrischen Funktionen gilt eine zusätzliche Genauigkeitsgrenze von $\pm 1 \times 10^{-9}$ für das *angezeigte Ergebnis*.

RECHENBEREICH

Zur Erhöhung der Rechengenauigkeit arbeitet Ihr HP-65 intern immer mit einer zehnstelligen Zahl und einem Exponentialfaktor. Diese verkürzte Zahlenschreibweise wird wissenschaftliche Schreibweise (oder Exponentialdarstellung) genannt. So

entspricht beispielsweise 23712.45 in wissenschaftlicher Schreibweise der Zahl 2.371245×10^4 .

UNTERLAUF

Wenn ein Ergebnis zu klein ist, um in ein Register gespeichert werden zu können ($< 10^{-99}$), wird das Register mit Null besetzt (und ein gegebenenfalls laufende Programm angehalten).

ÜBERLAUF

Falls das Ergebnis einer Berechnung die Kapazität eines Register überschreitet ($> 10^{99}$), wird es als Zahl angezeigt, die aus lauter Neunen besteht. Ein gegebenenfalls laufendes Programm wird angehalten.

TEMPERATURBEREICH

Betrieb:	0° bis +40° C (32° bis 104° F)
Laden der Batterien:	+10° bis +40° C (50° bis 104° F)
Lagerung:	-40° bis +55° C (-40° bis 131° F)

Der Temperaturbereich für das Schreiben von Magnetkarten ist 5° bis 40° C (41° bis 104° F). Bei Temperaturen unter 10° C (50° F) ist zu beachten, daß hier der Transport der Magnetkarten schon so verlangsamt sein kann (d.h. die Information auf der Karte so dicht), daß ein Lesen dieser Magnetkarten bei normalen Temperaturen nicht mehr fehlerfrei möglich ist.

UMWANDLUNG VON ZAHLEN

Wenn Sie eine Oktalzahl in die entsprechende dezimale Form umrechnen wollen, muß die Ausgangsgröße eine gültige Oktalzahl sein. Eine ungültige Oktalzahl (d.h. eine Zahl, die eine 8 oder 9 enthält) wird in eine falsche Dezimalzahl umgewandelt, ohne daß dabei eine Fehleranzeige (Blinken des Anzeigeregisters) auftritt.

ZUBEHÖR

Zum Bestellen von Zubehörteilen (auch Standardzubehör) für Ihren HP-65 verwenden Sie bitte die Bestellkarte, die Sie in der Tasche des hinteren Umschlags dieses Handbuches finden.

Richten Sie Ihre Bestellung an Ihr nächstgelegenes HP-Büro:

Für Deutschland:

Hewlett-Packard GmbH/Vertrieb

6000 Frankfurt 56

Berner Straße 117

Telefon (0611) 50 04-1

Für die Schweiz

Hewlett-Packard (Schweiz) AG

Zürcherstraße 20

Postfach 64

8952 Schlieren-Zürich

Telefon (01) 98 18 21 und 98 52 40

Für Österreich/

Für sozialistische Staaten und UdSSR:

Hewlett-Packard Ges.m.b.H.

Handelskai 52/3

Postfach 7

A-1205 Wien, Österreich

Telefon (0222) 33 66 06 bis 09

STANDARD-ZUBEHÖR

Ihr HP-65 wird Ihnen mit jeweils einem der folgenden Zubehörteile geliefert:

Zubehör	Modell/Teile Nr.
Aufladbare Batterie	82001A
Ladegerät (115/230 V)	82010A
Reisekassette	82018A
Weichledertasche	82017A
HP-65 Bedienungs-Handbuch	00065-90220
Namensschilder (4 Stück)	7120-2946
HP-65 Kurzanleitung	00065-90225
Standard-Paket bestehend aus:	00065-67013
• Anleitungsbuch	
• unbeschriebene Taschen-Anweisungskarten (20)	
• beschriebene Magnetkarten (19)	
• Magnetkopf-Reinigungskarte	
• unbeschriebene Magnetkarten (20)	
Block mit Programm-Arbeitsblättern	9320-0616

ZUSÄTZLICHES ZUBEHÖR

Weitere Zubehörteile und die verschiedenen erhältlichen Programmpakete sind auf der Bestellkarte in der Klappe des hinteren Umschlags aufgeführt. Dazu gehören:

Zubehör	Modell/Teile-Nr.
Batterie-Halter mit Batterien	82004A
Sicherheitsgehäuse	82015A
Bereitschaftstasche	82016A
unbeschriebene Programmkarten (40)	00065-67010
unbeschriebene Programmkarten (100)	00065-67032
Block mit Programm-Arbeitsblättern	9320-0616
unbeschriebene Taschen-Anweisungskarten (20)	9320-0613

Der HP-82004A-Batterie-Halter mit Batterien besteht aus einem Ladeanschluß und einem weiteren Batteriesatz. Auf diese Weise können Sie jeweils eine Batterie laden, während Sie die andere benutzen.

PFLEGE UND WARTUNG

RECHNER-PRÜFVERFAHREN

HINWEIS

Laden Sie die Batterie, bevor Sie den Rechner netz-unabhängig verwenden wollen.

Ein aufladbarer Batteriesatz wurde Ihnen mit Ihrem Rechner geliefert. Sie können den Rechner während des Ladens der Batterien verwenden; achten Sie aber darauf, daß er volle 14 Stunden am Ladegerät angeschlossen war, bevor Sie ihn unterwegs mitnehmen. Es ist für den Ladevorgang unbedeutend, ob Sie den Rechner dabei in Betrieb nehmen oder ausschalten.

VORSICHT

Starke statische Aufladungen können den Rechner beschädigen.

BATTERIESPANNUNG LÄSST NACH

Alle Dezimalpunkte leuchten auf und zeigen Ihnen damit an, daß nur noch etwa 2 bis 5 Minuten Betriebszeit verbleiben. Sie müssen dann entweder:

1. das Gerät am Netz betreiben
2. den Batteriesatz laden
3. einen geladenen Batteriesatz einsetzen.

KEINE ANZEIGE

Wenn die Anzeige ausbleibt, schalten Sie den HP-65 aus, bringen Sie den W/PRGM-RUN-Schalter in Stellung RUN und

schalten Sie den Rechner erneut ein. Wenn jetzt nicht 0.00 in der Anzeige erscheint, prüfen Sie:

1. ob der Batteriesatz vielleicht entladen ist oder keinen richtigen Kontakt mit dem Rechner hat;
2. ob der Rechner am Netz einwandfrei arbeitet;
3. ob der Netzstecker des Ladegerätes (falls Sie es angeschlossen haben) in einer unter Spannung stehenden Steckdose steckt.
4. Falls Sie immer noch keine Anzeige erhalten, ist der Rechner defekt (beachten Sie den Abschnitt Garantie).

GARANTIE

Die Garantie auf Ihren HP-65 erstreckt sich auf Material- und Verarbeitungsfehler und wird für 1 Jahr nach Auslieferung gewährt. Dabei werden fehlerhafte Teile instandgesetzt oder ausgetauscht, wenn der Rechner nach den unten angegebenen Anweisungen an Hewlett-Packard eingesandt wird.

Die Garantie erstreckt sich nicht auf solche Schäden, die durch Gewalteinwirkung entstanden sind oder auf Reparatur oder Veränderungen durch dritte zurückzuführen sind.

Weitergehende Ansprüche können nicht geltend gemacht werden. Hewlett-Packard haftet nicht für etwaige Folgeschäden.

NACH ABLAUF DER GARANTIEFRIST

Auch nach der einjährigen Garantiezeit können Sie Ihren HP-65 jederzeit zur Reparatur an uns einsenden, dann aber gegen eine geringe Berechnung. Senden Sie uns den Rechner zusammen mit Batteriesatz, Ladegerät und Reisekassette (beachten Sie die Versandanweisungen). Falls nur die Batterie defekt ist, bestellen Sie bitte eine Ersatzbatterie (siehe Bestellkarte in der Klappe des hinteren Umschlags).

VERSANDANWEISUNGEN

Bei fehlerhaftem Arbeiten des Rechners oder Ladegerätes senden Sie uns:

1. Ihren HP-65 mit Batteriesatz, Ladegerät und Reisekassette.
2. Eine komplett ausgefüllte Servicekarte (aus der Klappe des hinteren Umschlags).

Ist nur die Batterie defekt und die Garantiefrist noch nicht abgelaufen, senden Sie uns:

1. Nur den defekten Batteriesatz.
2. Eine komplett ausgefüllte Servicekarte (aus der Klappe des hinteren Umschlags).

Senden Sie den Rechner an die nächstliegende der auf der Servicekarte angegebenen Anschriften und achten Sie auf eine ausreichende und sichere Verpackung der Teile. In der Regel wird Ihr Rechner innerhalb von fünf Arbeitstagen nach Erhalt repariert und an Sie zurückgesandt. Sollten weitere servicebezogene Fragen auftreten, so rufen Sie eine der auf der Servicekarte angegebenen Telefonnummern an.

LADEN DER BATTERIEN UND NETZBETRIEB

Zur Vermeidung von Spannungsspitzen sollten Sie Ihren HP-65 ausschalten, bevor Sie das Ladegerät an den Rechner anschließen. Sie können ihn wieder einschalten, nachdem der Stecker des Ladegerätes am Netz angeschlossen worden ist. Während des Ladevorgangs können Sie Ihren HP-65 benutzen.

Das Laden einer vollständig entladenen Batterie dauert 14 Stunden. Kürzere Ladezeiten haben auch eine entsprechend kürzere Betriebszeit des Batteriesatzes zur Folge. Es ist sehr zweckmäßig, die Batterien über Nacht zu laden.

Ihr HP-65 kann auf Wunsch auch dauernd am Netz angeschlossen bleiben, ohne daß die Batterien darunter leiden. Sie sollten aber auch dann den Batteriesatz nicht aus dem

Rechner entfernen, da ein Betrieb der Karten-Lese/Schreib-Einrichtung nur in Verbindung mit dem Batteriesatz möglich ist. Wenn die Batterien vollständig entladen waren, sind sie zuerst für mindestens fünf Minuten zu laden, bevor eine Magnetkarte beschrieben oder gelesen werden kann. Das Aufleuchten sämtlicher Dezimalpunkte während des Transportvorgangs einer Programmkarte ist ein Zeichen dafür, daß der Batteriesatz geladen werden muß.

VORSICHT

Der Betrieb des HP-65 am Netz ohne eingesetzten Batteriesatz kann Beschädigungen am Rechner verursachen.

Verfahren Sie zum Laden des Batteriesatzes wie folgt:

1. Überzeugen Sie sich, daß die am Ladegerät eingestellte Spannung mit Ihrer Netzspannung übereinstimmt.

VORSICHT

Ihr HP-65 kann beschädigt werden, wenn Sie ihn an das Ladegerät anschließen und dieses nicht auf die richtige Netzspannung eingestellt ist.

2. Schalten Sie den Rechner aus.
3. Stecken Sie den Ladestecker in die rückwärtige Buchse am Rechner und den Netzstecker des Ladegerätes in eine Steckdose.
4. Schalten Sie den Rechner ein. Falls der W/PRGM-RUN-Schalter in der Stellung RUN steht, sollten Sie 0.00 in der Anzeige sehen.
5. Schalten Sie den Rechner aus, wenn Sie ihn während der Dauer des Ladens nicht benutzen wollen.
6. Nach Abschluß des Ladevorganges können Sie Ihren HP-65 weiterhin am Netz betreiben oder nach 7. verfahren.
7. Schalten Sie den Rechner aus und trennen Sie das Ladegerät vom HP-65 und vom Netz.

VORSICHT

Ihr Rechner kann beschädigt werden, falls Sie ein anderes Ladegerät verwenden, als das mitgelieferte Ladegerät HP 82010A.

BATTERIEBETRIEB

Verwenden Sie stets nur die aufladbaren Batterien HP 82001. Mit einem vollständig geladenen Batteriesatz können Sie den Rechner ca. 3 Stunden lang ununterbrochen benutzen. Wenn Sie das Gerät immer dann abschalten, wenn Sie es nicht benötigen, wird die Kapazität der Batterien leicht für einen ganzen Arbeitstag ausreichen.

Sämtliche Dezimalpunkte leuchten in der Anzeige auf, wenn Ihnen noch 2 bis 5 Minuten Betriebszeit verbleiben. Auch in diesem Fall ist die richtige Stellung des Dezimalpunktes eindeutig erkennbar, da für ihn eine ganze Anzeigestelle verwendet wird.

AUSTAUSCHEN DES BATTERIESATZES

Wenn Sie den Batteriesatz wechseln wollen, verfahren Sie nach folgenden Anweisungen:

1. Schalten Sie den Rechner aus und ziehen Sie das Ladekabel ab.
2. Schieben Sie die Riegel des Batteriefach-Deckels nach unten.

3. Drehen Sie den Rechner nach Abnehmen des Deckels um und lassen Sie den Batteriesatz herausfallen.



4. Kontrollieren Sie die mechanische Spannung der beiden Batteriekontakte und biegen Sie sie gegebenenfalls etwas nach oben, um einen einwandfreien Kontakt mit dem Batteriesatz zu gewährleisten.



5. Setzen Sie die neue Batterie ein (achten Sie auf die richtige Lage der Kontakte) und setzen Sie den Deckel auf.





6. Batteriedeckel unter den unteren Rand des Gehäuses schieben.



7. Verschließen Sie das Batteriefach, indem Sie die beiden Riegel wieder nach oben schieben.

ANMERKUNG

Wenn Sie den HP-65 häufig im Gelände oder auf Reisen aus Batterien betreiben, sind Sie sicher an dem Batteriehalter- und Lader HP 82004A interessiert, der aus einem besonderen Ladeadapter und einem zusätzlichen Batteriesatz besteht. Sie können dann jeweils eine Batterie laden, während Sie den Rechner mit der zweiten Batterie netz-unabhängig betreiben.

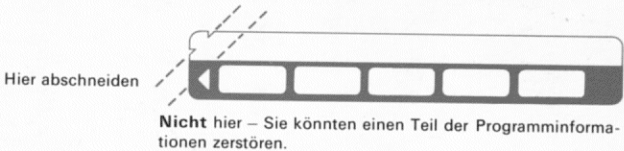
Es gehört zu den spezifischen Eigenschaften von Nickel-Kadmium-Zellen, daß ihre Kapazität bisweilen etwas nachläßt. Wenn Sie dies am Nachlassen der Betriebszeit feststellen, schalten Sie den HP-65 für mindestens 5 Stunden ein, womit die Batterie tiefentladen wird. Wenn Sie daran einen mindestens 14stündigen Ladevorgang anschließen, sollte der Batteriesatz wieder seine alte Kapazität haben.

Wenn sich die Batterien in kurzer Zeit von selbst entladen, sind sie wahrscheinlich defekt. Kommt dies innerhalb der Garantiefrist vor, so senden Sie den Batteriesatz unter Beachtung der bereits genannten Versandanweisungen an Hewlett-Packard ein. Ist die Garantie bereits abgelaufen, so bestellen Sie einen neuen Batteriesatz (unter Verwendung der Bestellkarte in der Klappe des hinteren Umschlags).

MAGNETKARTEN

SCHÜTZEN EINER MAGNETKARTE

Sie können eine Magnetkarte gegen ein versehentliches Überschreiben mit anderen Informationen schützen, indem Sie – wie gezeigt – den markierten Eckenabschnitt entfernen.



AUFBEWAHRUNG UND PFLEGE

Versuchen Sie, die Magnetkarten so weit wie möglich von Öl, Fett und Staub fernzuhalten. Schmutzige Magnetkarten können die Funktion der Lese/Schreib-Einrichtung Ihres Rechners nur beeinträchtigen. Sie können die Karten mit Alkohol (Spiritus) und/oder einem weichen feuchten Tuch reinigen.

Ihren Rechner schützen Sie vor staubiger und schmutziger Umgebung am besten, wenn Sie ihn stets in dem mitgelieferten Weichlederetuis aufbewahren.

124 Pflege und Wartung

Jeder Satz Magnetkarten enthält eine Reinigungskarte für den Magnetkopf der Karten-Lese/Schreib-Einrichtung.

ABRASIVE CARD FOR CLEANING RECORDING HEAD
CONSULT MANUAL FOR RECOMMENDED USE
— THIS SIDE UP —



Der Magnetkopf ähnelt denen bei Tonband- und Videorecordern. Auch hier können Ansammlungen von Staub und Fremdkörper auf dem Kopf den Kontakt zu der Magnetschicht der Programmkarten beeinträchtigen, was zu Fehlern beim Lesen und Schreiben der Karten führen kann. Die Reinigungskarte hat eine raue Oberfläche und dient dazu, diese Verunreinigungen zu entfernen. Wird sie allerdings übertrieben oft und grundlos (d.h. bei sauberem Magnetkopf) verwendet, kann dies die Lebensdauer des Kopfes nachteilig beeinflussen, da sie dann eine winzige Menge des Kopfmateri als abträgt. Nichtsdestoweniger sollten Sie sie ruhig verwenden, wenn Sie Verschmutzungen des Magnetkopfes vermuten. Wenn ein ein- bis fünfmaliger Durchlauf der Reinigungskarte den Fehler nicht behebt, sollten Sie allerdings Ihren Rechner zum Service einschicken.

BESCHRIFTEN DER MAGNETKARTEN

Die nicht-magnetisierbare Seite der Karten können Sie mit allen Schreibgeräten beschriften, die die Karte nicht beschädigen. Außer dem Programmtitel, den Sie oben auf die Karte schreiben können, ist es oft sehr zweckmäßig, die jeweiligen Bedeutungen und Funktionen der Tasten **A**, ..., **E** in die Kästchen an der Unterseite der Karte zu schreiben.

Es ist nicht zu empfehlen, die Karten mit Schreibmaschine zu beschriften, da dies (durch die Einprägungen) zu Funktionsfehlern führen kann.

VERWENDUNG DER ZWEITEN MAGNETSPUR

Sie können ein Programm auch auf der gegenüberliegenden Spur aufzeichnen (und später lesen), wenn Sie dabei die Karte

gegen die Pfeilrichtung in den Rechner einführen. Auf diese Weise kann eine Karte sogar zwei 100-Schritt-Programme fassen. Wir empfehlen Ihnen allerdings die Verwendung einer getrennten Magnetkarte für jedes Programm, da

1. das zweite Programm nicht ohne weiteres beschriftet werden kann;
2. größte Vorsicht geboten ist, wenn Sie diese Programmspur schützen wollen (schneiden Sie nicht mehr ab, als auf der gegenüberliegenden Seite, sonst zerstören Sie vielleicht bereits Programminformationen);
3. der Antriebsmechanismus mit seiner Transportwalze über der zweiten Spur liegt und so nach einiger Zeit Lesefehler auftreten können.

FEHLERHAFTE FUNKTION DER KARTEN-LESE/SCHREIB-EINRICHTUNG

Wenn Ihr Rechner offensichtlich fehlerfrei arbeitet und nur im Zusammenhang mit dem Schreiben und Lesen von Magnetkarten Schwierigkeiten auftauchen:

1. Überzeugen Sie sich davon, daß der W/PRGM-RUN-Schalter in der gewünschten Stellung steht: W/PRGM zum Aufzeichnen und RUN zum Einlesen von Programmen.
2. Falls der Transportmotor beim Einführen einer Programmkarte nicht anläuft, prüfen Sie, ob der Batteriesatz einwandfreien Kontakt mit dem Rechner hat und ausreichend geladen ist. Denken Sie daran, daß das Ladegerät alleine nicht in der Lage ist, den Strombedarf des Kartenlesers zu decken. Eine schon weitgehend entladene Batterie müssen Sie zum Betrieb des Kartenmotors zusammen mit dem Netzgerät verwenden. War die Batterie vollständig entladen, dann warten Sie 5 Minuten, bevor Sie den Kartenleser verwenden.
3. Falls der Karten-Antriebsmechanismus einwandfrei arbeitet, ein fehlerfreies Schreiben oder Lesen von Programmkarten aber dennoch nicht möglich ist, liegt der Fehler sicherlich in der Verschmutzung der Aufnahme-/Wieder-

gabe-Köpfe. Lassen Sie dann, wie angegeben, die Reinigungskarte einmal durch die Lese-/Schreib-Station laufen. Testen Sie anschliessend den Rechner unter Verwendung der mitgelieferten Diagnostik-Programme, und beachten Sie dabei die dazugehörigen Anweisungen. Wenn der Fehler nicht beseitigt ist, sollten Sie Ihren Rechner an eine der HP-Niederlassungen einsenden oder ihn selbst vorbeibringen.

WARNUNG

Die Magnetkarten können versehentlich gelöscht werden, wenn sie extrem starken Magnetfeldern ausgesetzt werden. Die Metallsuchgeräte auf Flughäfen stellen keine Gefahr dar.

ANHANG D

TYPISCHE FEHLER

1. Das Vorhandensein mehrerer gleicher Marken in benutzer-eigenen Funktionen im Programmspeicher. Kann auftreten, wenn vergessen wurde, vor Eintasten der Funktionen im W/PRGM-Modus mit **f** **PRGM** den Programmspeicher zu löschen.
2. Versehentliches Löschen eines gespeicherten Programms, wenn beim Einführen der Programmkarte der W/PRGM-RUN-Schalter in Stellung RUN steht.
3. Versehentliches Löschen eines Programms auf Magnetkarte, wenn die ungeschützte Karte im W/PRGM-Modus eingeführt wurde.
4. Eintasten unerwünschter Programmschritte in den Speicher, wenn der W/PRGM-RUN-Schalter versehentlich in Stellung W/PRGM stand.
5. Sie haben vergessen, die zu einer gewünschten Funktion gehörende Präfixtaste zu drücken (**■**, **f** oder **f¹**).
6. Verlust des Inhalts des T-Registers, da eine Neueingabe oder der Rückruf eines Wertes aus einem Speicherregister den Stack nach oben verschoben haben.
7. Der Inhalt von Register R₉ ist zerstört, da eine trigonometrische Funktion, eine Testoperation oder eine Umwandlung von rechtwinkligen in Polarkoordinaten ausgeführt wurde.
8. Fälschlicherweise nicht berücksichtigt, daß ein kombinierter Code vorlag und ein **NOP** in einem Sprung über zwei Speicherpositionen als Lückenfüller nötig gewesen wäre.
9. Ausführen trigonometrischer Funktionen im falschen Winkel-Modus.

128 Typische Fehler

10. Versuch, mit **[0]**, ..., **[9]** markierte Funktionen aufzurufen. Nur Funktionen, die mit **[A]**, ..., **[E]** markiert wurden, können über diese Tasten aufgerufen werden.
11. Sie haben vergessen, Flags vor ihrer Benutzung zu löschen.
12. Nach Aufruf einer benutzereigenen Funktion vom Tastenfeld aus (Buchstaben **[A]**, ..., **[E]**) oder durch ein Programm erwarten Sie fälschlicherweise unveränderte Inhalte der verschiedenen Register.
13. Verwendung von **[DSZ]** innerhalb eines Programms ohne bei der Programmführung R_8 mit dem entsprechenden Wert zu besetzen.
14. Berechnung von $f(x, y)$ mit versehentlich vertauschtem x und y .
15. Versehentliches Löschen von Programm und Daten durch Ausschalten des Rechners, Herausziehen des Ladekabels oder Einstecken während des Betriebes.
16. Fehler im Programmablauf, verursacht durch unerlaubtes Ineinanderschachteln von Programmteilen. Für die Schachtelung von Unterprogrammen gilt: Ein Programm kann eine Funktion aufrufen, die dann allerdings ihrerseits die Kontrolle an das rufende Programm zurückgeben muß, bevor eine andere Funktion aufgerufen werden kann. Sehen Sie sich die Beschreibung von **[RTN]** in Abschnitt 4 an.
17. Sie haben mit **[CLX]** Null in das **X**-Register gespeichert und nicht bedacht, daß **X** vom nächsten Wert überschrieben wird, da **[CLX]** den Stack-Lift verhindert. Verwenden Sie statt **[CLX]** die Taste **[0]**.

STICHWORTVERZEICHNIS

A

- A B C D E** oberste Tastenreihe 5
- Addition 6
 - Grad, Minuten, Sekunden 23
 - Inhalte der Speicherregister 30
 - Inhalte der Stackregister 6
- Adressierbare Register 29
 - Registerarithmetik 32
 - Speichern und Abruf 29
- Antilogarithmus 48
 - Basis 10 48
 - Basis e 48
- Anzeige 19
 - blinkende 21, 23
 - Batterie schwach 116
 - bei Programmausführung 22
- Anzeigeformat 20
- Arcus Cosinus 40
- Arcus Sinus 40
- Arcus Tangens 40
- Arithmetische Operationen
 - Stackregister 32
 - Speicherregister 32
- Austausch der Inhalte von X und Y 8
- Automatische Verschiebung der Stackinhalte 8

B

- Batterie
 - Aufladung 118
 - Austausch 120
 - Betrieb 120
- Bedingte Sprünge 83

130 Stichwortverzeichnis

Blinkende Anzeige **19**

Bogenmaß → Altgrad, Neugrad **41**

C

CHS Vorzeichenumkehr **6**

CLX Löschen des X-Registers **19**

COS Cosinus **40**

Code

 Programm **56**

 Tasten **56**

D

DEG Altgrad **41**

DEL Löschen von Programmschriften **68**

+DMS Dezimale Winkel → Grad, Minuten, Sekunden **40**

DMS+ Addition/Subtr. von Grad, Minuten, Sekunden **40**

DSP siehe Anzeigeformate **20, 41**

DSZ Dekrementieren und Sprung bei Null **88**

Definieren einer Marke **73**

Dekrement und Sprung bei Null **88**

Dezimalzahlen ↔ Oktalzahlen **45**

Direkte Programmsprünge **73**

Division **6**

E

EEX Exponenteneingabe **24**

ENTER↵ Inhalt von X in Y duplizieren **8**

e^x

Eingabe von Zahlen **24**

Eingabefehler **19, 23**

Einlesen von Magnetkarten **10**

Einlesen von Programmen **10**

Einschalter **5**

Exponentialfunktionen **47**

F

f **f⁻¹** Umschalttasten **7**

Fehleranzeige **23**

Fehler, häufige **108, 127**
 Fehlersuche **109**
 Fertige Programme **10**
 Festkomma-Anzeige **20**
 Flags **91**
 Funktionen **38**
 absoluter Wert **46**
 Addition von Grad, Minuten, Sekunden **23**
 Cosinus **40**
 dezimale Winkel \rightarrow Grad, Minuten, Sekunden **40**
 Dezimalzahlen \rightarrow Oktalzahlen **46**
 Exponentialfunktionen **47**
 Flags **91**
 Ganzzahlen **46**
 Kehrwert **46**
 ln **48**
 log **48**
 n-Fakultät **48**
 programmierbare Funktionen **53**
 Quadratwurzeln **48**
 rechtwinklige Koordinaten \leftrightarrow Polarkoordinaten **40**
 sinus **40**
 tangens **40**
 Test Flag **91**
 Winkelfunktionen **38, 40**

G

GRD Neugrad **41**
GTO go to **65, 83**
 Ganzzahliger Anteil **46**
 Gebrochener Anteil **46**
 Genauigkeit **111**
 Gewährleistung **117**
 Gleitkomma **21**
 Grad, Minuten, Sekunden **41**
 Größer als **84**

I

INT Ganzzahl/gebrochener Anteil **46**

K

- Kehrwert 48
- Kettenrechnungen 9
- Kleiner als 83
- Kombinierte Codes 58
- Koordinaten-Umwandlung 40
- Komma 22
- Korrektur eines Programms 70

L

- LBL** Marke 73
- LN** In 48
- LOG** log 48
- LST X** letzter x-Wert 25
- Lese-/Schreiboperationen 63
- Letzter x-Wert 25
- Logischer Vergleich 84
- Löschen
 - Flags 91
 - Programm 19, 60
 - Programmschritte 68
 - Register 19
 - alle Register 19
 - Stackregister 19
 - Vortasten 18
 - X-Register 19

M

- Magnetkarten 63
 - einlesen 10
 - registrieren 63
 - reinigen 123
 - schützen 64
- Magnetkopf reinigen 124
- Marken 73
- Multiplikation 6

N

NOP keine Operation 109
Negative Exponenten 25
Negative Zahlen 6
n-Fakultät 48

O

→OCT Umwandlung in Oktalzahlen 45

P

PREFIX Löschen von Vorwahltasten 18
PRGM Programmspeicher löschen 60
Polarkoordinaten ↔ rechtwinklige Koordinaten 40
Programm 52
 ausführen 62
 berichtigen 65, 70, 106
 Definition 52, 105
 Eingabe mit Magnetkarten 11
 Eingabe über Tastenfeld 53
 Fehlersuche 109
 Formulare 115
 Registrieren auf Magnetkarte 15
 Schreiben 13
 Schritt-Anzeiger 55
 Schritt einfügen 66
 Schritt löschen 68
 Speicher 54
 Sprünge 72
 Schutz 64
 Überprüfung 98
 mit Unterprogrammen 81
 Verzweigungen 72

Q

Quadratzahlen 40
Quadratwurzeln 40

R

-
- R↑** Verschieben der Stackinhalte nach oben 33
 - R↓** Verschieben der Stackinhalte nach unten 33
 - RAD** Bogenmaß 41
 - RCL** Rückruf 29
 - REG** Löschen der adressierbaren Speicher 19
 - R→P** rechtwinklige → Polarkoordinaten 40
 - R/S** Programm anhalten, ausführen 96
 - RTN** Programmspeicher auf Anfang stellen 53
 - Rechenbereich 112
 - Register 29
 - Registerarithmetik 32
 - Registrieren eines Programms 63
 - Rechtwinklige Koordinaten → Polarkoordinaten 40
 - Rückruf aus Speicherregistern 29
 - Runden der Anzeige 20

S

-
- SF1** **SF2** setzen von Flag 1, setzen von Flag 2 92
 - SIN** Sinus 23
 - SST** nächster Programmschritt 55
 - STK** Stackregister löschen 19
 - STO** Speichern 29
 - Schleifen 83
 - Schützen einer Magnetkarte 64
 - Service 117
 - Speicherregister 29
 - Stackregister 7, 33
 - Stoppen eines Programms 96
 - Subtraktion 6
 - Grad, Minuten, Sekunden 23
 - Speicherregister 30
 - Stackregister 6

T

-
- TAN** Tangens 40
 - TF1** **TF2** Test Flag 1, Test Flag 2 91

Tastencode **56**
 Tastenfeld **5**
 Temperaturbereich **113**
 T-Register **8**
 Trigonometrische Funktionen **40**

U

Überlauf **21,113**
 Überspringen von Programmschritten **72**
 Umkehrfunktionen **40, 48**
 Umwandlung **45**
 Dezimalzahlen → Oktalzahlen **45**
 dezimale Winkel → Grad, Minuten, Sekunden **41**
 rechtwinklige → Polarkoordinaten **40**
 Unterprogramme **77**
 Unterprogramme, Sprung zu **77**



V

Vergleichsoperationen **83**
 Verschieben von Werten **8**
 Verzweigungen **88**
 Vorzeichen **6**

W

W/PRGM-RUN-Schalter **12**
 in Stellung: W/PRGM, Erstellung und Korrektur von
 Programmen, Registrieren von Programmen auf
 Magnetkarten
 in Stellung: RUN, zum Einlesen von Programmen, für
 Berechnungen über Tastenfeld, Ausführung
 gespeicherter Programme
 Wartung **116**
 Winkelfunktionen **39**
 Winkel-Modus **41**

X

 Kehrwert **48**
 Quadratwurzel **48**

136 Stichwortverzeichnis

$x \leftrightarrow y$ Austausch der x- und y-Werte 8
 $x \neq y$ $x \leq y$ $x = y$ $x > y$ logische Vergleiche 83
X-Register 8

Y

y^x Exponentialfunktionen 47
Y-Register 8

Z

Zahlen
 Anzeige 6
 Eingabe 6, 96
 verschieben 8
Z-Register 8
Zubehör 114



172 mal Verkauf und Service in 65 Ländern

Hewlett-Packard GmbH/Vertrieb:

6000 Frankfurt 56, Berner Straße 117, Postfach 560140, Telefon (0611) 50 04 -1

3000 Hannover-Kleefeld, Mellendorfer Straße 3, Telefon (0511) 55 60 46

2000 Hamburg 1, Wendenstrasse 23, Telefon (040) 24 13 93

8500 Nürnberg, Hersbrückstraße 42, Telefon (0911) 57 10 66/75

8012 Ottobrunn, Isar Center, Unterhachinger Straße 28, Telefon (089) 601 30 61/67

7030 Böblingen, Herrenbergerstraße 110, Telefon (07031) 66 72 87

4000 Düsseldorf, Vogelsanger Weg 38, Telefon (0211) 63 80 31/38

1000 Berlin 30, Keith Straße 2-4, Telefon (030) 24 90 86

Für die Schweiz: Hewlett-Packard (Schweiz) AG, Zürcherstraße 20, Postfach 64,
8952 Schlieren-Zürich, Telefon (01) 98 18 21 und 98 52 41

Für Österreich/Für sozialistische Staaten und UdSSR:

Hewlett-Packard Ges.m.b.H., Handelskai 52/53, Postfach 7, A-1205 Wien,
Österreich, Telefon (0222) 33 66 06 bis 09

Europa-Zentrale:

Hewlett-Packard S.A., 7, rue du Bois-du-Lan, Postfach 349, CH-1217 Meyrin 1-Genf,
Schweiz, Telefon (022) 41 54 00

Scan Copyright ©
The Museum of HP Calculators
www.hpmuseum.org

Original content used with permission.

Thank you for supporting the Museum of HP
Calculators by purchasing this Scan!

Please to not make copies of this scan or
make it available on file sharing services.