



Elektronische Taschenrechner

BEDIENUNGSANLEITUNG



(Programmanleitung für PR56D-NC
Liegt dem Gerät bei)

Bestell-Nr. PR 56 D-NC 463.3335
SR 58 D-NC 462.9135
SR 60 D-NC 463.2337

BEDIENUNGSANLEITUNG

privileg SR58D-NC
SR60D-NC
PR56D-NC
Elektronische Taschenrechner

INHALTSVERZEICHNIS

1. EINLEITUNG
2. STROMVERSORGUNG
3. EINSCHALTEN DES RECHNERS
4. DAS BEDIENUNGSFELD
Bedeutung der Tasten und Schalter
5. ZAHLENEINGABE UND ANZEIGE
Wahl des Anzeigeformats und der Nachkommastellen
Eingabe positiver und negativer Fließkommazahlen
Wissenschaftliche Zahlendarstellung
Eingabe der Zahl Pi
Korrektur von Eingabefehlern Umwandlung der Anzeigeformate
6. GRUNDRECHENARTEN
Addition und Subtraktion mit Fehlerkorrektur
Multiplikation und Division mit Fehlerkorrektur
7. DIE FUNKTIONSTASTEN
Kehrwert $1/x$
Quadratwurzel \sqrt{x}
Potenz zur Basis e e^x
Potenz zur Basis 10 10^x
Potenz zu einer beliebigen Basis y^x
Wurzel mit beliebigem Wurzelexponenten $x\sqrt[y]{}$
Quadrat x^2
Natürlicher Logarithmus $\ln x$
Logarithmus zur Basis 10 $\log x$
Logarithmus zu einer beliebigen Basis $\log_a b$
Trigonometrische Funktionen \sin, \cos, \tan
Trigonometrische Umkehrfunktionen $\arcsin, \arccos, \arctan$
Hyperbelfunktionen \sinh, \cosh, \tanh
Hyperbel-Umkehrfunktionen $\operatorname{arsinh}, \operatorname{arcosh}, \operatorname{artanh}$
8. DER KONSTANTENSPEICHER
Speichern einer Zahl Speicherabruf
Speicher-Addition und Subtraktion
Speicher-Multiplikation und Division

9. DIE KLAMMERN

10. ZUSAMMENGESetzte RECHNUNGEN

- Verkettete Klammern
- Summe von Produkten
- Produkt von Summen
- Rechnen mit Funktionen
- Registeraustausch

11. ANZEIGE VON KAPAZITÄTSÜBERLAUF UND FEHLER

12. RECHENGENAUIGKEIT UND RECHENGESCHWINDIGKEIT

13. ANWENDUNGSBEISPIELE

- Freier Fall
- Kugelvolumen
- Berechnungen an dünnen Linsen
- Flächenträgheitsmoment
- Statistik
- Zinseszinsrechnung

14. SPEICHERORGANISATION

1. EINLEITUNG

Die Modelle "privileg SR 58/SR 60/PR 56 D-NC" zählen zu einer Serie leistungsfähiger wissenschaftlicher Rechner modernster Konzeption. Ihre Elektronik wurde in der MOS/LSI - Technik gefertigt. Die MOS - Technik (Metal Oxide Semiconductor) gestattet die Herstellung kleinster Transistoren, Dioden und Widerstände mit sehr geringem Energiebedarf. Diese elektronischen Bauelemente werden in einem Arbeitsgang zu tausenden auf einem Halbleiterplättchen aufgebaut. Dadurch passt die gesamte Logik, das Herz des Rechners, auf eine Fläche von $10 \times 20 \text{ mm}^2$ Diese dichte Packung von Bauelementen, die das geringe Volumen der Rechner ermöglicht, nennt man LS I, Large Scale Integration.

Neben dem hohen technischen Leistungsstandard wurde bei der Konstruktion dieser Rechnerserie selbstverständlich auch auf vielseitige Einsatzmöglichkeit und leichte Bedienung Rücksicht genommen. Die große Anzahl technisch wissenschaftlicher Funktionen, Klammer-automatik und rechnende Speicher ermöglichen einfache Problemlösung. Mit dem programmierbaren Modell haben Sie Zusätzlich die Möglichkeit, einen einmal eingetasteten Rechenablauf beliebig oft abzurufen.

Damit Sie Ihren Rechner gründlich kennenlernen, haben wir die mit vielen Beispielen und Rechentips versehene Bedienungsanleitung zusammengestellt. Lesen Sie bitte die folgenden Seiten! Der Zeitaufwand dafür bringt Ihnen beim Gebrauch des Rechners wiederum viel Zeitersparnis.

2. STROMVERSORGUNG

Ihr Rechner wird betriebsfertig mit eingebauten aufladbaren Nickel Cadmium-Batterien und Ladegerät geliefert. Die Batterien sind beim fabrikneuen Gerät der besseren Haltbarkeit wegen nicht geladen. Aus diesem Grund sollte der Rechner vor der ersten Inbetriebnahme 14-16 Stunden geladen werden.

LADEVORGANG

- 1) Rechner ausschalten.
- 2) Ladekabel in den Anschluss an die Oberseite des Gerätes stecken.
- 3) Ladegerät an 220 Volt Wechselstrom anschließen.
- 4) Völlig entladene Batterien 14-16 Stunden aufladen; teilweise entladene Batterien entsprechend kürzer. Die Batterien sollten mit Rücksicht auf ihre Lebensdauer nicht zu stark überladen werden. Während des Ladevorgangs kann gerechnet werden, Die Ladezeit wird dadurch etwa verdoppelt.
- 5) Falls während des Ladens gerechnet wurde, Rechner ausschalten.
- 6) Ladegerät vom Netz trennen.
- 7) Ladekabel vom Rechner trennen.

Bei schwachen Batterien leuchtet links in der Anzeige die Marke \perp auf.

Bitte laden Sie die Batterien rechtzeitig nach. Entladene Batterien können zu falschen Rechenergebnissen führen. Die Lebensdauer der Batterien beträgt etwa vier Jahre. Wegen eines eventuell notwendigen Batteriewechsels wenden Sie sich bitte an den nächsten Quelle-Kundendienst.

3. EINSCHALTEN DES RECHNERS

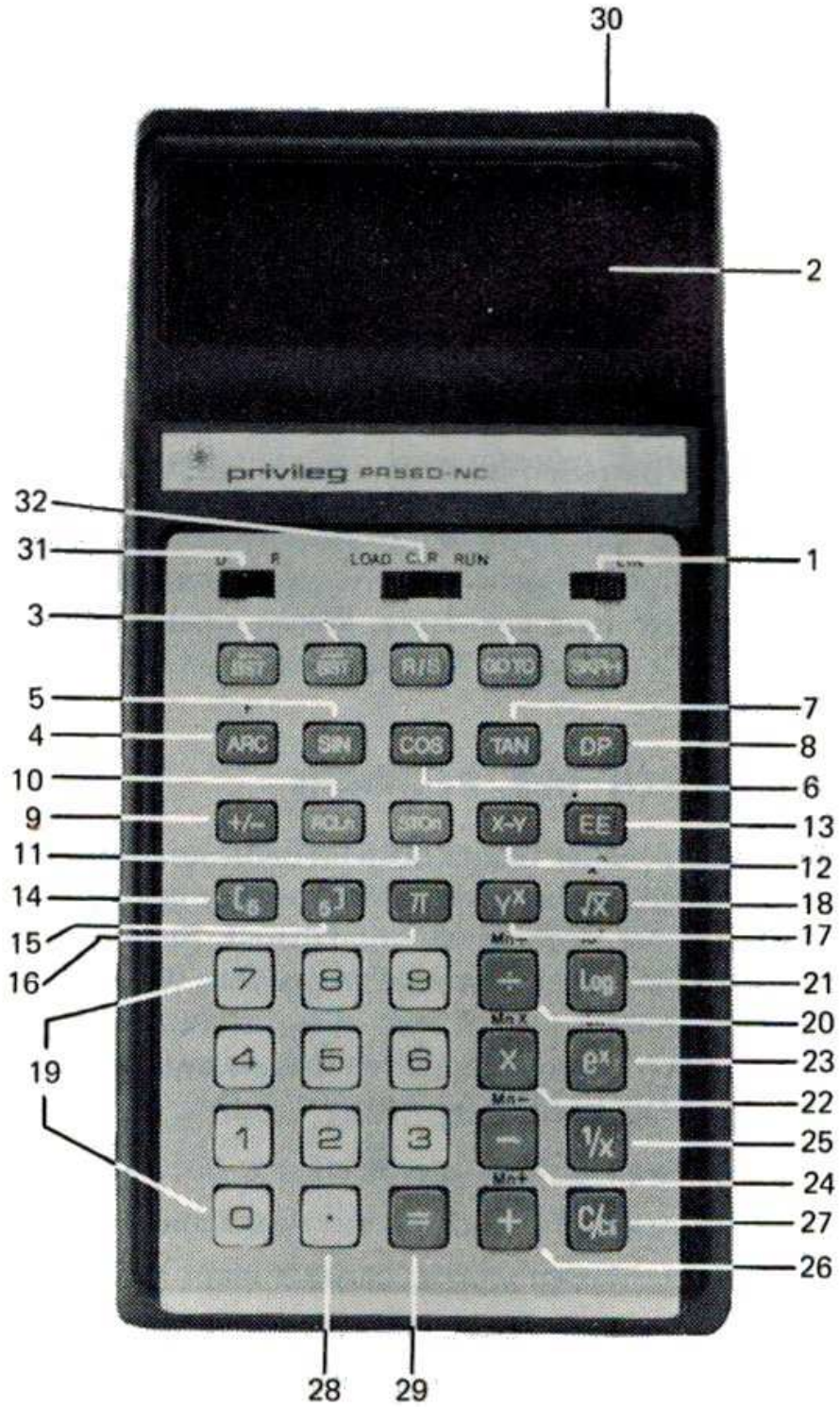
Der EIN/AUS-Schalter rechts oben im Bedienungsfeld wird in Stellung "EIN" geschoben. Dabei werden alle Speicher gelöscht. In der Anzeige erscheint 0.00 der Rechner ist einsatzbereit. Beim Einschalten werden automatisch zwei Nachkommastellen gewählt. Wünscht man eine davon abweichende Anzahl, kann man diese eintasten. Näheres dazu im Kapitel 5. Beim umprogrammierten Rechnen muss der Schiebeschalter des PR 56 D-NC in Stellung "RUN" stehen.

Wenn im weiteren Betrieb die Rechnung mit $\boxed{=}$, oder $\boxed{=}$ gefolgt von einem beliebigen Tastendruck mit Ausnahme der Grundrechenarten und 0 abgeschlossen wurde, braucht nicht gelöscht zu werden. Ist man sich dessen nicht sicher, drückt man vorsorglich zweimal die rote $\boxed{C/CE}$ Taste. Man vermeidet dadurch Fehler, die entstehen können, wenn versehentlich vor Beginn der eigentlichen Rechnung schon Daten eingegeben wurden.

Sollte ausnahmsweise beim Einschalten keine **0.00** sondern eine Reihe von Zeichen in der Anzeige stehen, schalten Sie bitte aus und nach vier Sekunden wieder ein.

Ihr privileg - Rechner auf einen Blick

Gezeigt ist das Modell PR 56D-NC. Bei den Modellen SR 58D-NC und SR 60D-NC fehlen die Programmsteuertasten.



- 1) Ein/Aus-Schalter
- 2) Anzeige
- 3) Programmsteuertasten
- 4) Doppelfunktionstaste
- 5) Sinus/Arkussinus
- 6) Cosinus/Arkuscosinus
- 7) Tangens/Arkustangens
- 8) Format-Wahltaste
- 9) Vorzeichenwechsel-Taste
- 10) Speicherabruf-Taste
- 11) Speicher-Taste
- 12) Austauschaste Arbeitsspeicher x Arbeitsspeicher y
- 13) Eingabe des Zehnerexponenten
- 14) Taste für das Öffnen der Klammern
- 15) Taste für das Schließen der Klammern
- 16) Fest Zahl Pi
- 17) Allgemeine Exponentialfunktion
- 18) Quadratwurzel/Quadrat
- 19) Zifferntasten
- 20) Division/Division der Zahl im Speicher durch die Zahl in der Anzeige
- 21) Dekadischer Logarithmus/Potenz zur Basis 10
- 22) Multiplikation/Multiplikation der Zahl im Speicher mit der Zahl in der Anzeige
- 23) Potenz zur Basis e/Natürlicher Logarithmus
- 24) Subtraktion/Subtraktion der Zahl in der Anzeige von der gespeicherten Zahl
- 25) Kehrwert
- 26) Addition/Addition der Zahl in der Anzeige zur gespeicherten Zahl
- 27) Anzeigelöschung/Gesamtlöschung
- 28) Kommaeingabe
- 29) Ergebnistaste
- 30) Buchse für Netzkabel
- 31) "D - R" Schiebeschalter zur Wahl der Einheit der Winkel
- 32) Schiebeschalter für programmiertes Rechnen

4. DAS BEDIENUNGSFELD

Das Bedienungsfeld besteht aus 40 Druckpunkt-Tasten, dem Ein/Aus-Schalter "EIN" und dem "D-R" Schalter, der festlegt in welcher Einheit Winkel verarbeitet werden. Die Ziffern und Symbole stehen auf den Tasten. Bei 10 Tasten kann mit Hilfe von $\boxed{F/ARC}$ eine zweite Funktion abgerufen werden. Das Symbol dieser Funktion steht über der betreffenden Taste.

Zusätzlich besitzt die programmierbare Version noch 5 Programm-Steuertasten.

ZIFFERN-TASTEN	$\boxed{0}$ bis $\boxed{9}$ Durch Tastendruck bringt man die betreffende Ziffer in die Anzeige
DEZIMALPUNKT	$\boxed{\bullet}$ Diese Taste bestimmt die Lage des Dezimalpunktes.
GRUNDRECHENARTEN	$\boxed{+}$ $\boxed{-}$ $\boxed{\times}$ $\boxed{\div}$ Diese Tasten teilen dem Rechner mit, welche Rechenoperation mit der nächsten eingegebenen Zahl ausgeführt werden soll. Gleichzeitig werden Zwischenergebnisse angezeigt.
ERGEBNIS-TASTE	$\boxed{=}$ Diese Taste bringt das Ergebnis der vorhergegangenen Rechnung in die Anzeige.
VORZEICHEN-WECHSEL	$\boxed{+/-}$ Der Druck auf diese Taste ändert das Vorzeichen der in der Anzeige stehenden Zahl. Will man einen negativen Zahlenwert, so gibt man zuerst die Zahl ein und drückt anschließend die $\boxed{+/-}$ Taste. Das negative Vorzeichen erscheint links der eingegebenen Zahl. Auch zur Eingabe negativer Zehner-exponenten dient die $\boxed{+/-}$ Taste.
EINLESEN DES ZEHNEREXPONENTEN	\boxed{EE} Mit dieser Taste gibt man bei der wissenschaftlichen Zahlendarstellung den Zehner-exponenten ein.

KREISZAHL PI

π

Diese Taste bringt die Kreiszahl $\pi = 3.1415927$ in die Anzeige.

DATENAUSTAUSCH

$x \leftrightarrow y$

Diese Taste tauscht die Zahl in der Anzeige (x-Register) gegen die Zahl im Arbeits-speicher (y-Register) aus.

KLAMMER TASTEN

$()$ (oder $[]$ für SR60/PR56)

Die Klammertasten weisen den Rechner an, den von ihnen eingeschlossenen Ausdruck unabhängig von der vorausgegangenen Eingabe zu berechnen und in den Klammerregistern zu speichern. Nach dem Schließen der Klammer steht in der Anzeige der Wert dieses Ausdrucks. Klammern dürfen geschachtelt werden. da der SR 58 zwei, die Modelle SR 60 und PR 56 fünf Klammerregister besitzen.

LOSCHTASTE

C/CE

Diese Taste hat drei Funktionen:

1. Einmaliges Drücken der C/CE Taste bringt den Rechner nach einer Fehleranzeige, die durch eine unzulässige Operation verursacht wurde, wieder in Arbeitsbereitschaft. Alle Speicher bis auf den Konstanten-speicher M werden gelöscht.
2. Einmaliges Drücken der C/CE Taste nach einer Zahleneingabe löscht allein die Anzeige. Alle weiteren Informationen bleiben gespeichert.
3. Zweimaliges Betätigen dieser Taste löscht alle im Rechner gespeicherten Daten mit Ausnahme der Werte im Konstanten-speicher.

FORMAT WAHL

\boxed{DP}

Mit dieser Taste wird die Anzahl der Nachkommastellen und die wissenschaftliche Zahlendarstellung gewählt.

KEHRWERT

$\boxed{1/x}$

Diese Taste weist den Rechner an, den Kehrwert der in der Anzeige stehenden Zahl zu berechnen.

QUADRATWURZEL

$\boxed{\sqrt{x}}$

Diese Taste weist den Rechner an, die Quadratwurzel aus der in der Anzeige stehenden Zahl zu ziehen.

FUNKTIONSTASTE

\boxed{F} \boxed{ARC}

Im Anschluss an die Betätigung dieser Taste gelten nicht mehr die auf den Tasten stehenden Funktionen sondern die über den Tasten stehenden Angaben und die Arkusfunktionen. Ein Punkt links im Anzeigefeld zeigt dies an. Zur übersichtlicheren Darstellung wird im Anschluss an die Bedienung von \boxed{F} immer die über der gedrückten Taste stehende Funktion geschrieben. Versehentliches Bedienen von \boxed{F} wird durch nochmaligen Druck rückgängig gemacht.

QUADRIERTASTE

$\boxed{x^2}$

Diese Taste weist den Rechner an, die in der Anzeige stehende Zahl zu Quadrieren.

POTENZ ZUR BASIS e

$\boxed{e^x}$

Diese Taste weist den Rechner an, den Wert $e = 2.71828$ in die Potenz x zu erheben. wobei der Exponent x die in der Anzeige stehende Zahl ist.

POTENZ ZUR BASIS 10

$\boxed{10^x}$

Diese Taste weist den Rechner an, die Potenz 10^x zu berechnen, x ist die Zahl in der Anzeige.

NATURLICHER LOGARITHMUS	$\boxed{F} \boxed{Ln}$ Diese Taste weist den Rechner an, den natürlichen Logarithmus I Logarithmus zur Basis $e = 2.7182818$ der in der Anzeige stehenden Zahl zu ermitteln.
LOGARITHMUS ZUR BASIS 10	\boxed{Log} Diese Taste weist den Rechner an, den Logarithmus zur Basis 10 der in der Anzeige stehenden Zahl zu ermitteln.
EXPONENTIAL FUNKTION	$\boxed{y^x}$ Diese Taste weist den Rechner an, die Basis y (zuerst eingegebene Zahl) in die Potenz x (zweite eingegebene Zahl) zu erheben.
SPEICHER-TASTE	\boxed{STO} (oder \boxed{STOn}) Diese Taste weist den Rechner an, die angezeigte Zahl zu speichern. Der bis dahin im Speicher stehende Wert wird vorher automatisch gelöscht.
SPEICHER ABRUFTASTE	\boxed{RCL} (oder \boxed{RCLn}) Diese Taste weist den Rechner an, den gespeicherten Wert anzuzeigen. Der Speicher-inhalt wird dabei nicht verändert.
SPEICHER MULTIPLIKATION	$\boxed{F} \boxed{M \times}$ (oder $\boxed{F} \boxed{Mn \times}$) Diese Taste weist den Rechner an, die in der Anzeige stehende Zahl mit dem Inhalt des Konstantenspeichers zu multiplizieren. Das Produkt steht im Konstantenspeicher, die Anzeige wird nicht verändert.
SPEICHER DIVISION	$\boxed{F} \boxed{M \div}$ (oder $\boxed{F} \boxed{Mn \div}$) Diese Taste weist den Rechner an, den Inhalt des Konstantenspeichers durch die in der Anzeige stehende Zahl zu dividieren. Der Quotient steht im Konstantenspeicher, die Anzeige wird nicht verändert.
SPEICHER ADDITION	$\boxed{F} \boxed{M +}$ (oder $\boxed{F} \boxed{Mn +}$) Diese Taste weist den Rechner an, die in der Anzeige stehende Zahl zum Inhalt des Konstantenspeichers zu addieren. Die Summe steht im Konstantenspeicher, die Anzeige bleibt unverändert.

SPEICHER SUBTRAKTION

$\boxed{F} \boxed{M-}$ (oder $\boxed{F} \boxed{Mn-}$)

Diese Taste weist den Rechner an, die in der Anzeige stehende Zahl vom Inhalt des Konstantenspeichers zu subtrahieren. Die Differenz steht im Konstantenspeicher, die Anzeige bleibt unverändert.

D-R SCHALTER

Die Stellung dieses Schiebeschalters legt fest, in welcher Einheit die Winkel bei der Berechnung trigonometrischer Funktionen verarbeitet werden.

D bedeutet Grad (Altgrad)

R bedeutet Radiant (rad, Bogenmaß)

TRIGONOMETRISCHE FUNKTIONEN

\boxed{SIN} , \boxed{COS} , \boxed{TAN}

Diese Tasten veranlassen die Berechnung der sin-, cos- oder tan-Werte des in der Anzeige stehenden Winkels.

TRIGONOMETRISCHE UMKEHR-FUNKTIONEN

$\boxed{ARC} \boxed{SIN}$, $\boxed{ARC} \boxed{COS}$, $\boxed{ARC} \boxed{TAN}$

Diese Tasten veranlassen die Berechnung der arcsin- arccos- oder arctan-Werte der in der Anzeige stehenden Zahl.

Die Tasten zur Programmsteuerung des PR 56 NC sind im Kapitel "Programmieren" beschrieben. (liegt dem Gerät separat bei).

5. ZAHLENEINGABE UND ANZEIGE

Die Leuchtanzeige hat folgenden Aufbau:



WAHL DES ANZEIGE FORMATS UND DER NACHKOMMASTELLEN.

WICHTIG - BITTE BEACHTEN!

Nachstehender Hinweis wurde bei allen Beispielen in dieser Anleitung berücksichtigt, (also Eintasten **DP** 4).

Mit der Taste **DP** gefolgt von einer Ziffer zwischen 0 und 7 wählt man die Fließkommadarstellung einer Zahl einschließlich der gewünschten Zahl von Nachkommastellen.

DP 8 oder **DP** 7 schaltet auf die wissenschaftliche Zahlen-darstellung um.

Der Rechner arbeitet intern in jedem Fall mit 10 Stellen, sodass immer eine hohe Rechengenauigkeit gewährleistet ist. Das einmal eingestellte Format bleibt erhalten bis ein anderes eingetastet oder der Rechner ausgeschaltet wird.

EINGABE POSITIVER UND NEGATIVER FLIESSKOMMAZAHLEN

Eine Zahl von maximal acht Stellen wird einschließlich des Dezimalpunktes so eingetastet, wie man sie schreibt. Tastet man mehr als acht Ziffern ein, so werden die Überzähligen nicht beachtet. Eine Null vor dem Komma darf weggelassen werden. Will man einen negativen Zahlenwert, so drückt man **nach** der Zahleneingabe die **+/-** Taste. Das negative Vorzeichen erscheint links der Mantisse.

Übersteigt das Ergebnis einer Rechnung 8 Stellen, so wird automatisch die wissenschaftliche Zahlendarstellung gewählt.

WISSENSCHAFTLICHE ZAHLENDARSTELLUNG UMWANDLUNG DER ANZEIGE FORMATE

Mit Hilfe des zweiteiligen Exponenten rechts in der Anzeige kennen Zahlen zwischen $1 \cdot 10^{-99}$ und $9.9999999 \cdot 10^{99}$ eingegeben werden.

Nach Eintasten der maximal achtstelligen Mantisse drückt man \boxed{EE} und gibt den Zehnerexponenten ein. Negative Exponenten erhält man durch Betätigen der $\boxed{+/-}$ Taste vor oder nach Eingabe der Exponenten-ziffern. Das Vorzeichen erscheint links vom Exponenten.

Die Mantisse 1 muss man nicht eintasten. Man beginnt sofort mit \boxed{EE} .

EINGABE DER ZAHL PI

Die Zahl $\pi = 3.1415927$ wird durch einfachen Tastendruck auf die Taste $\boxed{\pi}$ in die Anzeige gebracht. Eine vorher dort stehende Zahl wird überschrieben, muss also nicht gelöscht werden. (Vorher \boxed{DP} $\boxed{8}$ drücken, um achtstellige Anzeige zu bewirken).

KORREKTUR VON EINGABEFEHLERN

Fehlerhafte Zahleneingabe löscht man durch einmaliges Bedienen der $\boxed{C/CE}$ Taste. Die übrigen vorher eingegebenen Informationen werden dadurch nicht verändert.

Beispiel: $2 \times 4 = 8$

Eingabe	Anzeige	Kommentar
2	2	
$\boxed{\times}$	2	
3	3	Fehlerhafte Zahleneingabe!
$\boxed{C/CE}$	0	Die Eingabe 3 wurde gelöscht; Die Eingabe 2 $\boxed{\times}$ bleibt gespeichert.
4	4	Richtiger Faktor.
$\boxed{=}$	8	Richtiges Ergebnis,

UMWANDLUNG DER ANZEIGEFORMATE

Eingabe	Anzeige	Kommentar
Einschalten	0.00	Beim Einschalten werden automatisch 2 Nachkommastellen gewählt.
12.3456	12.3456	Zahleneingabe
<input type="checkbox"/> DP <input type="checkbox"/> 3	12.346	3 Nachkommastellen. Es wird gerundet.
<input type="checkbox"/> DP <input type="checkbox"/> 1	12.3	1 Nachkommastelle
<input type="checkbox"/> DP <input type="checkbox"/> 8	1.2345600 01	Abruf der wissenschaftlichen Zahlendarstellung.

6. GRUNDRECHENARTEN

Es ist leicht, die Tasten und Schalterfunktionen Ihres Rechners zu beherrschen. Die folgenden Seiten zeigen Ihnen anhand von Beispielen die Bedienung. Arbeiten Sie bitte die Beispiele mit Ihrem eigenen Rechner durch!

Achten Sie darauf, dass zu Beginn jeder Rechnung alle Speicher gelöscht sind. Gerät dazu kurz Aus und wieder "EIN"-schalten.

Bei den folgenden Beispielen werden der besseren Übersicht wegen die vom Rechner gezeigten Füllnullen weggelassen.

Wenn nicht anders angegeben, wählt man 4 Nachkommastellen, DP 4 . Der Schiebeschalter des PR 56 D·NC muss in Stellung "RUN" stehen.

ADDITION UND SUBTRAKTION

Beispiel 1: 5+3=8

Eingabe	Anzeige
5	5
<input type="checkbox"/> +	5
3	3
<input type="checkbox"/> =	8

Beispiel 2: $5-3=2$

Eingabe	Anzeige
5	5
<input type="text" value="-"/>	5
3	3
<input "="" type="text" value="="/>	2

Beispiel 3: $55.755 - 108.71 = -52.955$

Eingabe	Anzeige	Kommentar
55.755	55.755	
<input type="text" value="-"/>	55.755	
108.71	108.71	
<input "="" type="text" value="="/>	-52.955	Das Vorzeichen links der Mantisse zeigt das negative Ergebnis an.

Die Tasten für die Grundrechenarten schließen die vorausgegangene Rechnung ab. Das Zwischenergebnis steht in der Anzeige. Die Taste zum Beispiel hat die gleiche Funktion wie die Tastenkombination . Das sieht man an folgendem Beispiel:

Beispiel 4: $2-6+9=5$

Eingabe	Anzeige	Kommentar
2	2	
<input type="button" value="-"/>	2	
6	6	
<input type="button" value="+"/>	-4	Die <input type="button" value="+"/> Taste hat die Rechnung $2 - 6$ abgeschlossen und das Zwischenergebnis -4 in die Anzeige gebracht. Es ist nicht notwendig, die <input type="button" value="="/> Taste zu bedienen.
9	9	
<input type="button" value="="/>	5	

Beispiel 5: $2 \cdot 10^7 + 3 \cdot 10^{10} = 3.002 \cdot 10^{10}$

Eingabe	Anzeige	Kommentar
2	2	
<input type="button" value="EE"/>	2	00 Aktivierung der Exponenteneingabe.
7	2	07 Eingabe des Zehnerexponenten.
<input type="button" value="+"/>	20000000	Da der Wert kleiner 10^8 ist, wird die in wissenschaftlicher Darstellung eingegebene Zahl automatisch in die Fließkommazahl umgewandelt.
3	3	
<input type="button" value="EE"/>	3	00
10	3	10
<input type="button" value="="/>	3.002	10 $3.002 \cdot 10^{10}$

FEHLER KORREKTUR

Wird bei einer umfangreichen Rechnung versehentlich die $\boxed{+}$ oder die $\boxed{-}$ Taste gedrückt, muss man nicht von vorne beginnen. Man kann die fehlerhafte Eingabe durch Addition oder Subtraktion einer Null, also mit der Tastenfolge $0 \boxed{=}$ korrigieren.

Beispiel: $5 - 3 = 2$

Eingabe	Anzeige	Kommentar
5	5	
$\boxed{+}$	5	Fehlerhafte Eingabe!
0	0	Korrektur durch Addition von Null
$\boxed{=}$	5	
$\boxed{-}$	5	Richtige Eingabe
3	3	
$\boxed{=}$	2	

MULTIPLIKATION UND DIVISION

Beispiel 1: $42 \times 5.31 = 22.302$

Eingabe	Anzeige
4.2	4.2
<input type="button" value="×"/>	4.2
5.31	5.31
<input "="" type="button" value="="/>	22.302

Beispiel 2: $22302 \div 0.4 = 55.755$

Eingabe	Anzeige	Kommentar
22.302	22302	
<input type="button" value="÷"/>	22.302	
.4	0.4	
<input "="" type="button" value="="/>	55.755	Die Null muss nicht eingetastet werden!

Beispiel 3: $2.3 \times 13.57 \div 6.89 = 4.5299$

Eingabe	Anzeige	Kommentar
2.3	2.3	
<input type="button" value="×"/>	2.3	
13.57	13.57	
<input type="button" value="÷"/>	31.211	Die <input type="button" value="÷"/> Taste hat die Rechnung abgeschlossen und das Zwischenergebnis 31.211 in die Anzeige gebracht. Es ist nicht notwendig, die <input type="button" value="="/> Taste zu bedienen.
6.89	6.89	
<input "="" type="button" value="="/>	4.5299	

Beispiel 4: $2.3 \cdot 10^{-15} \times 13.57 \cdot 10^{17} \div 76.89 \cdot 10^{16} = 4.5299 \cdot 10^{-14}$

Eingabe	Anzeige	
2.3	2.3	
<input type="button" value="EE"/>	2.3	00
15	2.3	15
<input type="button" value="+/-"/>	2.3	-15
<input type="button" value="x"/>	2.3	-15
13.57	13.57	
<input type="button" value="EE"/>	13.57	00
17	13.57	17
<input type="button" value="÷"/>	3121.1	
6.89	6.89	
<input type="button" value="EE"/>	6.89	00
16	6.89	16
<input type="button" value="="/>	4.5299	-14

FEHLERKORREKTUR

Wird versehentlich oder gedrückt, so kann man die falsche Dateneingabe mit der Tastenfolge 1 korrigieren.

Beispiel: $5-2=3$

Eingabe	Anzeige	Kommentar
5	5	
<input type="button" value="x"/>	5	Fehlerhafte Eingabe!
1	1	Korrektur durch Multiplikation mit 1!
<input type="button" value="="/>	5	
<input type="button" value="-"/>	5	Richtige Eingabe!
2	2	
<input type="button" value="="/>	3	

7. DIE FUNKTIONSTASTEN

Die Befehle $\boxed{1/x}$, $\boxed{x^2}$, $\boxed{\sqrt{x}}$, $\boxed{e^x}$, $\boxed{10^x}$, $\boxed{\ln}$, $\boxed{\log}$, die trigonometrischen Funktionen und deren Umkehrfunktionen verwenden jeweils die Zahl in der Anzeige. Die betreffende Funktion wird in besonderen Rechenwerken ermittelt und das Ergebnis in die Anzeige geschrieben. Der Inhalt des y-Registers wird nicht verändert. Dadurch können alle Grundrechenarten und die allgemeine Exponentialfunktion y^x auch mit Funktionswerten ausgeführt werden.

Die Funktion y^x arbeitet ähnlich wie die Tasten für die Grundrechenarten und verwendet die Zahlen in x- und y-Register. Alle Funktionen dürfen beliebig in Klammern eingeschlossen werden.

Die Berechnung der Funktionswerte nimmt im Höchstfall wenige Sekunden in Anspruch. Während dieser Zeit ist die Anzeige dunkel und es dürfen keine Daten eingegeben werden.

KEHRWERT $1/x$, $x \neq 0$

$$\text{Beispiel 1: } -\frac{1}{3} = -0.3333$$

Eingabe	Anzeige
3	3
$\boxed{+/-}$	-3
$\boxed{1/x}$	-0.3333

POTENZ ZUR BASIS e e^x

Beispiel: $e^{-0.2}=0.8187$

Eingabe	Anzeige
.2	0.2
$\boxed{+/-}$	-0.2
$\boxed{e^x}$	0.8187

POTENZ ZUR BASIS 10 10^x

Beispiel: $10^{2.55} = 354.8134$

Eingabe	Anzeige
2.55	2.55
\boxed{F} $\boxed{10^x}$	354.8134

POTENZ ZU EINER BELIEBIGEN BASIS y^x, y > 0

Mit Hilfe dieser Taste berechnet man den Potenzwert y^x auf folgende Weise: Einrasten von y, Druck auf die $\boxed{y^x}$ Taste, Eintasten von x. Druck auf die $\boxed{=}$ Taste oder eine gleichwertige Taste. (Sh. Kapitel 6)

Beispiel 1: $2^{-3} = 0.125$

Eingabe	Anzeige
2	2
$\boxed{y^x}$	2
3	3
$\boxed{+/-}$	-3
$\boxed{=}$	0.125

Die Taste $\boxed{y^x}$ schließt ähnlich wie die Tasten für die Grundrechenarten die vorangegangene Rechnung ab, wirkt also wie die Tastenkombination $\boxed{=}$ $\boxed{y^x}$

Beispiel 2: $(3 + 1.7)^{2.5} + 2.11 = 50$

Eingabe	Anzeige	Kommentar
3	3	
<input type="button" value="+"/>		
1.7	1.7	
<input type="button" value="y<sup>x</sup>"/>	4.7	Die Taste <input type="button" value="y<sup>x</sup>"/> hat die Rechnung $3 + 1.7$ abgeschlossen und das Zwischenergebnis 4.7 in die Anzeige gebracht. Die <input type="button" value="="/> Taste braucht hier nicht gedrückt zu werden.
2.5	2.5	
<input type="button" value="+"/>	47.89	Die Taste <input type="button" value="+"/> ist der Taste <input type="button" value="="/> gleichwertig.
2.11	2.11	
<input type="button" value="="/>	50	

WURZEL MIT BELIEBIGEM WURZELEXPONENTEN $^x\sqrt{y}$, $y>0$

Wurzeln mit beliebigem Wurzelexponenten a werden nach der Beziehung $^a\sqrt{y} = y^{1/a}$, auf folgende Weise berechnet: Eintasten der Basis y . Druck auf die $\boxed{y^x}$ Taste. Eintasten des Wurzelexponenten. Druck auf die $\boxed{1/x}$ Taste. Das Ergebnis wird mit der $\boxed{=}$ Taste abgerufen.

Beispiel: $^7\sqrt{128} = 2$

Eingabe	Anzeige
128	128
$\boxed{y^x}$	128
7	7
$\boxed{1/x}$	0.1429
$\boxed{=}$	2

QUADRAT x^2

Beispiel: $(3 + 1.5^2)^2 = 27.5625$

Eingabe	Anzeige	Kommentar
3	3	
$\boxed{+}$	3	
1.5	1.5	
$\boxed{F} \boxed{x^2}$	2.25	Bei der Berechnung von 1.5^2 bleibt die Information $3 +$ erhalten.
$\boxed{=}$	5.25	Vor dem Quadrieren muss man das Zwischenergebnis ermitteln. Der Rechner würde sonst 2.25 ins Quadrat erheben.
$\boxed{F} \boxed{x^2}$	27.5625	

NATÜRLICHER LOGARITHMUS $\ln x$, $x > 0$

Beispiel: $\ln 44^3 = 3 \times \ln 44 = 11.3526$

Eingabe Anzeige

3 3

3

44 44

3.7842

11.3526

LOGARITHMUS ZUR BASIS 10 (DEKADISCHER, BRIGGSCHER LOGARITHMUS) $\log x$, $x > 0$

Beispiel: $\log 423 = 2.62631$

Eingabe Anzeige

423 423

2.6263

LOGARITHMUS ZU EINER BELIEBIGEN BASIS $\log_a b$, $a > 0$; $b > 0$

Den Logarithmus von b zur Basis a , also $\log_a b$ erhält man aus der Gleichung
 $\log_a b = \log b / \log a = \ln b / \ln a$

Beispiel: $\log_2 423 = 8.7245$

Eingabe Anzeige

423 423

6.0474

6.0474

2 2

0.6931

8.7245

TRIGONOMETRISCHE FUNKTIONEN **sin, cos, tan**

TRIGONOMETRISCHE UMKEHR FUNKTIONEN **arcsin, arccos, arctan**

Winkel können in Grad oder Radian (rad, Bogenmaß) verarbeitet werden. Der Wahlschalter befindet sich links unterhalb der Anzeige. Die Werte der trigonometrischen Funktionen werden auch für Winkel über 360° und für negative Winkel berechnet. Je weiter man sich allerdings von $\pm 360^\circ$ entfernt, desto geringer wird die Rechengenauigkeit.

SINUS sin

Beispiel: $\sin 45^\circ = 0,7071$

Schieben Sie den Wahlschalter in Stellung "D" (Grad)!

Eingabe Anzeige

45 45

SIN 0.7071

COSINUS cos

Beispiel: $1 + \cos 30^\circ = 1.8660$

Eingabe Anzeige

1 1

+ 1

30 30

COS 0.8660

= 1.8660

TANGENS tan

Beispiel: $\sqrt{\tan^2(\pi/8) + 1} = 1.08248$

Schieben Sie den Wahlschalter in Stellung "R"

Eingabe	Anzeige
π	3.1416
\div	3.1416
8	8
=	0.3927
TAN	0.4142
F x^2	0.1716
+	0.1716
1	1
=	1.1716
\sqrt{x}	1.0824

ARCUSSINUS arcsin x $-1 \leq x \leq +1$

Beispiel: $\arcsin 0.5 = 0.5236$

Wahlschalter in Stellung "R"

Eingabe	Anzeige
.5	0.5
ARC sin	0.5236

ARCUSCOSINUS arccos x $-1 \leq x \leq +1$

Beispiel: $\arccos(-0.2) = 101.537^\circ$

Wahlschalter in Stellung "D"

Eingabe	Anzeige
.2	0.2
+/-	-0.2
ARC cos	101.537

ARCUTANGENS arctan x

Beispiel: $7.45 + (\arctan 5.5)^2 = 9.3847$

Wahlschalter in Stellung "R"

Eingabe	Anzeige	Kommentar
5.5	5.5	Um die Verwendung von Klammern oder eines Speichers beim Quadrieren zu umgehen, beginnt man mit der Berechnung des arctan-Wertes

1.3909

1.9347

1.9347

7.45 7.45

9.3847

HYPERBELFUNKTIONEN sinh, cosh, tanh

Die drei Hyperbelfunktionen sind wie folgt definiert:

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

Beispiel: $\sinh 0.82 = 0.9150$

Eingabe	Anzeige
---------	---------

.82 0.82

2.2705

2.2705

.82 0.82

-0.82

0.4404

1.8301

2 2

0.9150

Damit man den x-Wert nicht zweimal eingeben muss, kann man ihn auch speichern (Siehe Kapitel 8).

HYPERBEL·UMKEHR FUNKTIONEN arsinh, arcosh, artanh

Die Definition der hyperbolischen Umkehrfunktionen lautet:

$$\operatorname{arsinh} x = \ln(x + \sqrt{x^2 + 1}); \operatorname{arcosh} x = \ln(x + \sqrt{x^2 - 1})$$

$$\operatorname{artanh} x = \frac{1}{2} \ln \frac{1+x}{1-x}$$

Beispiel 1: $\operatorname{arsinh} 0.8 = 0.7327$

Eingabe	Anzeige
8	0.8
$\boxed{F} \boxed{x^2}$	0.64
$\boxed{+}$	0.64
1	1
$\boxed{=}$	1.64
$\boxed{\sqrt{x}}$	1.2806
$\boxed{+}$	1.2806
.8	0.8
$\boxed{=}$	2.0806
$\boxed{F} \boxed{\ln}$	0.7327

Beispiel 2: $\operatorname{artanh} 0.8 = 1.0986$

Eingabe	Anzeige	Kommentar
1	1	
$\boxed{+}$	1	
.8	0.8	
$\boxed{\div}$	1.8	
$\boxed{(5}$	0	Der Nenner wird in eine Klammer eingeschlossen (Siehe Kapitel 9)
1	1	
.8	0.8	
$\boxed{5)}$	0.2	
$\boxed{=}$	9	
$\boxed{F} \boxed{\ln}$	2.1972	
$\boxed{\div}$	2.1972	
2	2	
$\boxed{=}$	1.0986	

8. DER KONSTANTENSPEICHER

8.1 DER KONSTANTENSPEICHER DES MODELLS SR 58D-NC

Der Rechner "privileg SR 58 D-NC" besitzt einen rechnenden Konstantenspeicher M, dessen Inhalt bei der Ausführung algebraischer oder wissenschaftlicher Berechnungen nicht beeinflusst wird. Dieser Speicher dient zur Aufnahme von Zahlenwerten, die innerhalb einer Rechnung mehrmals benötigt werden oder zur Ausführung von Kettenrechnungen.

SPEICHERN EINER ZAHL

Eine Zahl in der Anzeige wird mit Hilfe der Taste $\boxed{\text{STO n}}$ gespeichert. Der ursprünglich im Speicher stehende Wert wird überschrieben, der Speicher muss also nicht gelöscht werden.

Beispiel: Im Speicher steht die Zahl 5: es soll 9.81 gespeichert werden.

Eingabe	Anzeige	Speicher	Kommentar
5	5	beliebig	
$\boxed{\text{STO n}}$	5	5	5 wird gespeichert
9.81	9.81	5	
$\boxed{\text{STO n}}$	9.81	9.81	Die 5 im Speicher wurde überschrieben, 9.81 ist gespeichert.

SPEICHERABRUF

Die Zahl im Speicher wird mit Hilfe der Taste $\boxed{\text{RCL n}}$ in die Anzeige gerufen. Die ursprünglich in der Anzeige stehende Zahl wird überschrieben, geht also verloren.

Beispiel: $5.12 \div 4.1 = 1.2488$
 $5.12 \div 3.6 = 1.4222$

Eingabe	Anzeige	Speicher	Kommentar
5.12	5.12	beliebig	
$\boxed{\text{STO n}}$	5.12	5.12	
$\boxed{\div}$	5.12	5.12	
4.1	4.1	5.12	
$\boxed{=}$	1.2488	5.12	
$\boxed{\text{RCL n}}$	5.12	5.12	
$\boxed{\div}$	5.12	5.12	
3.6	3.6	5.12	
$\boxed{=}$	1.4222	5.12	

SPEICHER ADDITION UND SUBTRAKTION

Mit der Tastenkombination $\boxed{F} \boxed{M+}$ wird die Zahl in der Anzeige zum Speicherinhalt addiert; mit $\boxed{F} \boxed{M-}$ wird die Zahl in der Anzeige vom Speicherinhalt subtrahiert.

Beispiel: $\sqrt{3^2 + 4^2 - 2^2} = 4.5826$

Eingabe	Anzeige	Speicher
3 $\boxed{F} \boxed{x^2}$	9	
\boxed{STOn}	9	9
4 $\boxed{F} \boxed{x^2}$	16	9
$\boxed{F} \boxed{M+}$	16	25
2 $\boxed{F} \boxed{x^2}$	4	25
$\boxed{F} \boxed{M-}$	4	21
\boxed{RCLn}	21	21
$\boxed{\sqrt{x}}$	4.5826	21

SPEICHER MULTIPLIKATION UND DIVISION

Mit der Tastenkombination $\boxed{F} \boxed{M \times}$ wird der Wert im Speicher mit der Zahl in der Anzeige multipliziert; die Tastenkombination $\boxed{F} \boxed{M \div}$ dividiert den Speicherinhalt durch die Zahl in der Anzeige.

Beispiel 1: $(1 + \ln 2) (1 + \sqrt{3}) = 4.6258$

Eingabe	Anzeige	Speicher
1	1	0
$\boxed{+}$	1	0
2	2	0
$\boxed{F} \boxed{\text{Ln}}$	0.6931	0
$\boxed{=}$	1.6931	0
$\boxed{\text{STOn}}$	1.6931	1.6931
$\boxed{1}$	1	1.6931
$\boxed{+}$	1	1.6931
3	3	1.6931
$\boxed{\sqrt{x}}$	1.7321	1.6931
$\boxed{=}$	2.7321	1.6931
$\boxed{F} \boxed{M \times}$	2.7321	4.6258
$\boxed{\text{RCLn}}$	4.6258	4.6258

Beispiel 2: $\frac{1 + \cos 20^\circ}{1 - \cos 30^\circ} = 14.4781$

Wahlschalter in Stellung "D"

Eingabe	Anzeige	Speicher
1 $\boxed{+}$	1	----
20 $\boxed{\text{COS}}$	0.9397	----
$\boxed{=}$ $\boxed{\text{STOn}}$	1.9397	1.9397
1 $\boxed{-}$		1.9397
30 $\boxed{\text{COS}}$	0.866	1.9397
$\boxed{=}$	0.134	1.9397
$\boxed{F} \boxed{M \div}$	0.134	14.4781
$\boxed{\text{RCLn}}$	14.4781	14.4781

8.2 DIE KONSTANTENSPEICHER DER MODELLE SR 60 D-NC UND PR 56 D-NC

Diese Modelle besitzen 10 rechnende Speicher Mn. Der Inhalt dieser Speicher wird bei der Ausführung der allgemeinen Rechenoperationen nicht beeinflusst. Sie dienen der Aufnahme von Konstanten, Teilergebnissen und zur Ausführung von Zwischenrechnungen.

SPEICHERN EINER AHL

Eine Zahl in der Anzeige wird mit Hilfe der Taste **STO n** gefolgt von der Speichernummer **0** bis **9** gespeichert. Die ursprünglich in den Speichern stehenden Werte werden überschrieben. Die Speicher müssen also nicht gelöscht werden.

Beispiel: Die Zahl 5.1 im Speicher Nummer 6 soll von der Zahl 9.81 überschrieben werden.

Eingabe	Anzeige	Speicher 6	Kommentar
5.1	5.1	beliebig	
STO n 6	5.1	5.1	5.1 wird im Speicher 6 gespeichert.
9.81	9.81	5.1	
STO n 6	9.81	9.81	Die 5.1 im Speicher 6 wurde überschrieben; 9.81 ist gespeichert.

SPEICHERABRUF

Die Zahlen in den Speichern werden mit Hilfe der Taste **RCL n** gefolgt von der Speichernummer **0** bis **9** in die Anzeige gerufen.

Die ursprünglich in der Anzeige stehende Zahl wird überschrieben, geht also verloren.

Beispiel: $5.12 \div 4.1 = 1.2488$
 $5.12 \div 3.6 = 1.4222$

Eingabe	Anzeige	Speicher 6
5.12 STO n 6	5.12	5.12
÷ 4.1	4.1	5.12
=	1.2488	5.12
RCL n 6	5.12	5.12
÷ 3.6	3.6	5.12
=	1.4222	5.12

SPEICHER ADDITION UND SUBTRAKTION

Mit der Tastenkombination $\boxed{F} \boxed{Mn+}$, gefolgt von der Speicher nummer wird die Zahl in der Anzeige zum Inhalt des angewählten Speichers addiert; mit $\boxed{F} \boxed{Mn-} \boxed{0} \dots \boxed{9}$ wird die Zahl in der Anzeige vom Inhalt des angewählten Speichers subtrahiert.

Beispiel: $\sqrt{3^2 + 4^2 - 2^2} = 4.5826$ berechnet im Speicher 0

Eingabe	Anzeige	Speicher 0
3 $\boxed{F} \boxed{x^2}$	9	
$\boxed{STOn} \boxed{0}$	9	9
4 $\boxed{F} \boxed{x^2}$	16	9
$\boxed{F} \boxed{Mn+} \boxed{0}$	16	25
2 $\boxed{F} \boxed{x^2}$	4	25
$\boxed{F} \boxed{Mn+} \boxed{0}$	4	21
$\boxed{RCLn} \boxed{0}$	21	21
$\boxed{\sqrt{x}}$	4.5826	21

SPEICHER MULTIPLIKATION UND DIVISION

Mit der Tastenkombination $\boxed{F} \boxed{Mn \times} \boxed{0} \dots \boxed{9}$ wird der Wert im angewählten Speicher mit der Zahl in der Anzeige multipliziert. Die Tastenkombination $\boxed{F} \boxed{Mn \div} \boxed{0} \dots \boxed{9}$ dividiert den Inhalt des angewählten Speichers durch die Zahl in der Anzeige.

Beispiel 1: $(1 + \ln 2)(1 + \sqrt{3}) = 4.6258$ berechnet im Speicher 1

Eingabe	Anzeige	Speicher
1 $\boxed{+}$	1	----
2 $\boxed{F} \boxed{Ln}$	0.6931	----
$\boxed{=}$ $\boxed{STOn} \boxed{1}$	1.6931	1.6931
1 $\boxed{+}$	1	1.6931
3 $\boxed{\sqrt{x}}$	1.7321	1.6931
$\boxed{=}$	2.7321	1.6931
$\boxed{F} \boxed{Mn \times} \boxed{1}$	2.7321	4.6258
$\boxed{RCLn} \boxed{1}$	4.6258	4.6258

Beispiel 2:
$$\frac{2 \times 0.6231 + \sin 25.4^{\circ}}{2 \times 0.6231 - \sin 25.4^{\circ}} = 2.0497$$

Um das mehrmalige Eintasten der Zahlen zu sparen, speichert man 2×0.6231 in Speicher 1 und $\sin 25.4^{\circ}$ in Speicher 2

Eingabe	Anzeige	Speicher 0	Speicher 1
0.6321 $\boxed{STOn} \boxed{1}$	0.6231	0.6231	
2	2	0.6231	
$\boxed{F} \boxed{Mn \times} \boxed{1}$	2	1.2462	
25.4 \boxed{SIN}	0.4289	1.2462	
$\boxed{STOn} \boxed{2}$	0.4289	1.2462	0.4289
$\boxed{RCLn} \boxed{1} \boxed{+}$	1.2462	1.2462	0.4289
$\boxed{RCLn} \boxed{2} \boxed{\div}$	1.6751	1.2462	0.4289
\boxed{C}	0	1.2462	0.4289
$\boxed{RCLn} \boxed{1} \boxed{+}$	1.2462	1.2462	0.4289
$\boxed{RCLn} \boxed{2} \boxed{C}$	0.8173	1.2462	0.4289
$\boxed{=}$	2.0497	1.2462	0.4289

9. DIE KLAMMERN

Ihr Rechner hat fünf (zwei beim SR 58 D-NC) Klammerregister. Klammern können also geschachtelt werden. Beim Setzen der Klammern muss man darauf Rücksicht nehmen, dass der Rechner nicht Punkt vor Strich rechnet. Die Klammern werden also manchmal anders gesetzt als im algebraischen Ausdruck. In der üblichen Schreibweise lässt man das Malzeichen vor Klammern häufig weg. Bei einem Rechner darf das nicht praktiziert werden, da er sonst nicht weiß, welche Rechenoperation mit dem Klammerausdruck ausgeführt werden soll.

Beispiel 1: $2 \times 3 + (4 \times 5) = 26$

Eingabe	Anzeige	Kommentar
2	2	
<input type="button" value="×"/>	2	
3	3	
<input type="button" value="+"/>	6	Das erste Produkt braucht nicht eingeklammert zu werden, da der <input type="button" value="+"/> Befehl die Berechnung von 2×3 Die Information $6 +$ wird im ersten Klammerregister gespeichert.
<input type="button" value="(5"/>	0	
4	4	
<input type="button" value="×"/>	4	
5	5	
<input type="button" value=")"/>	20	Beim Schließen der Klammer wird der eingeklammerte Ausdruck berechnet, die Klammer wirkt also wie das <input type="button" value="="/> und der Inhalt des Klammerregisters wird ins y-Register geschoben.
<input type="button" value="="/>	26	Der Klammerausdruck wird zu der 6 im y-Register addiert.

Würde die Rechnung ohne Verwendung der Klammer eingetastet, so wäre das Ergebnis 50, da die Tasten für die Grundrechnungsarten die Berechnung des vorangehenden Ausdrucks veranlassen, also $2 \times 3 + 4 \times 5 = (2 \cdot 3 + 4) \cdot 5 = 50$

Beispiel 2: $1.5 \times 2 + 4 \cdot (5 - 2) = 15$ wird eingetastet wie
 $1.5 \times 2 + ([4 \times [(5 - 2)]]) =$

Eingabe	Anzeige
1.5	1.5
\times	1.5
2	2
$+$	3
$(5$	0
4	4
\times	4
$(5$	0
5	5
$-$	5
2	2
$)$	3
$)$	12
$=$	15

Wenn zwei Klammern geöffnet werden, müssen auch zwei geschlossene Klammern folgen.

Beispiel 3: $2 + (4 \sin 30^\circ + 7 \cos 60^\circ) \times 2 = 13$

Eingabe	Anzeige	Kommentar
2	2	
<input type="button" value="+"/>	2	
<input type="button" value="(5"/>	0	Der Ausdruck 2+ wird gespeichert
<input type="button" value="(5"/>	0	
4	4	
<input type="button" value="×"/>	4	
30	30	
<input type="button" value="SIN"/>	0.5	
<input type="button" value="5)"/>	2	Der Rechner hat $4 \sin 30^\circ$ berechnet.
<input type="button" value="+"/>	2	
<input type="button" value="(5"/>	0	Der Ausdruck 2+ wird gespeichert.
7	7	
<input type="button" value="×"/>	7	
60	60	
<input type="button" value="COS"/>	0.5	
<input type="button" value="5)"/>	3.5	Der Rechner hat $7 \cos 60^\circ$ berechnet.
<input type="button" value="×"/>	5.5	Die Summe der beiden Klammern wird mit <input type="button" value="x"/> abgerufen und angezeigt.
2	2	
<input type="button" value="5)"/>	11	5.5×2 wurde durch Schließen der Klammer berechnet.
<input "="" type="button" value="="/>	13	Der eingangs gespeicherte Ausdruck 2+ wird abgerufen.

Wenn man im Zweifel ist, ob und wo man Klammern setzen soll überlege man immer, wie der Rechner ohne Klammern weiter rechnen würde.

10. ZUSAMMENGESetzte RECHNUNGEN

Zur Berechnung umfangreicher Ausdrücke hat man drei Möglichkeiten:

- 1) Die günstige Speicherorganisation des Rechners erlaubt zum Beispiel die Berechnung verketteter Klammerausdrücke ohne weiteres Hilfsmittel. Dabei berücksichtige man immer, dass die Tasten $\boxed{+}$, $\boxed{-}$, $\boxed{\times}$, $\boxed{\div}$ und $\boxed{y^x}$ die vorhergegangene Rechnung abschließen. Sie haben also die gleiche Funktion, wie die Tastenkombination $\boxed{=}\boxed{+}$, $\boxed{=}\boxed{-}$, $\boxed{=}\boxed{\times}$, $\boxed{=}\boxed{\div}$ und $\boxed{=}\boxed{y^x}$.
- 2) Teile eines Ausdrucks können zur weiteren Verwendung gespeichert werden. Man kann zum Speicherinhalt Zahlen addieren und subtrahieren, man kann im Speicher multiplizieren und dividieren.
- 3) Teile des Ausdrucks können eingeklammert werden.

In allen drei Fällen werden Funktionswerte wie Zahlen behandelt. Die betreffende Funktionstaste entnimmt die Zahl der Anzeige, berechnet den Funktionswert und stellt ihn in die Anzeige zurück. Vorher eingegebene Informationen werden nicht verändert.

VERKETTETE KLAMMERN

$$(3 + 4) 2 - 6$$

Beispiel 1: ----- = 1.6
5

Eingabe	Anzeige	Kommentar
3	3	
$\boxed{+}$	3	
4	4	
$\boxed{\times}$	7	$\boxed{\times}$ hat die Berechnung von 3 + 4 ausgelöst.
2	2	
$\boxed{-}$	14	$\boxed{-}$ hat die Berechnung von 7 x 2 ausgelöst.
6	6	
$\boxed{\div}$	8	$\boxed{\div}$ hat die Berechnung von 14 - 6 ausgelöst.
5	5	
$\boxed{=}$	1.6	

SUMME VON PRODUKTEN

Beispiel: $2 \times 3 + 4 \times 5 - 6(7 - 4) = 8$

Eingabe	Anzeige	Kommentar
2 <input type="button" value="×"/>	2	Das erste Produkt muss nicht geklammert werden.
3 <input type="button" value="+"/> 6	6	
<input type="button" value="(5"/>	0	
4 <input type="button" value="×"/>	4	Zweites Produkt
5 <input type="button" value="5)"/>	20	
<input type="button" value="-"/>	26	
<input type="button" value="(5"/>	0	
7 <input type="button" value="-"/>	7	Um eine Doppelklammer zu vermeiden beginnt man hier zweckmäßigerweise mit der Berechnung von 7 - 4.
4 <input type="button" value="×"/>	3	
<input type="button" value="5)"/>	18	
<input type="button" value="="/>	8	

Diese Aufgabe lässt sich auch sehr einfach mit Hilfe eines rechnenden Speichers lösen.

PRODUKT VON SUMMEN

Beispiel: $(3 + 4) (2 + 8) = 70$

Eingabe	Anzeige	Kommentar
3	3	
<input type="text" value="+"/>	3	
4	4	
<input type="text" value="×"/>	7	Der Befehl für die Multiplikation der ersten mit der zweiten Klammer wird gesetzt.
<input type="text" value="(5"/>	0	Die Berechnung des zweiten Klammersausdrucks wird eingeleitet.
2	2	
<input type="text" value="+"/>	2	
8	8	
<input type="text" value=")"/>	10	Durch Schließen der Klammer wurde der zweite Ausdruck berechnet.
<input type="text" value="="/>	70	Die <input type="text" value="="/> Taste multipliziert die beiden Summen miteinander.

RECHNEN MIT FUNKTIONEN

Beispiel 1: $\ln(3 + \sqrt{5}) = 1.6556$

Eingabe	Anzeige	Kommentar
3	3	
<input type="text" value="+"/>	3	
5	8	
<input type="text" value="√x"/>	2.2361	Die 5 in der Anzeige wurde durch $\sqrt{5}$ ersetzt. 3 + ist weiterhin gespeichert.
<input type="text" value="="/>	5.2361	
<input type="text" value="F"/> <input type="text" value="Ln"/>	1.6556	

Beispiel 2: $\log(100 + e^5) = 2.3952$

Eingabe	Anzeige
---------	---------

100	100
-----	-----

$+$	100
-----	-----

5	5
---	---

e^x	148.4132
-------	----------

$=$	248.4132
-----	----------

Log	23952
-----	-------

$$e^{3 + \ln 2}$$

Beispiel 3: $\frac{e^{3 + \ln 2}}{4 + \sqrt{9}} = 5.7387$

Eingabe	Anzeige	Kommentar
---------	---------	-----------

3	3	
---	---	--

$+$	3	
-----	---	--

2	2	
---	---	--

F Ln	0.6931	
------	--------	--

$=$	3.6931	
-----	--------	--

e^x	40.1711	
-------	---------	--

\div	40.1711	
--------	---------	--

(5)	0	Der Nenner wird in Klammern gesetzt.
-----	---	--------------------------------------

4	4	
---	---	--

$+$	4	
-----	---	--

9	9	
---	---	--

\sqrt{x}	3	
------------	---	--

)5)	7	Durch Schließen der Klammer wurde der Nenner berechnet.
-----	---	---

$=$	5.7387	
-----	--------	--

sin 20°

Beispiel 4: $(\frac{\sin 20^\circ}{3} + 4 \cos 30^\circ + 5) \times 6.3 = 54.0421$

Wahlschalter in Stellung "D"

Eingabe	Anzeige
20	20
$\boxed{\text{SIN}}$	0.342
$\boxed{\div}$	0.342
3	3
$\boxed{+}$	0114
$\boxed{(5)}$	0
4 $\boxed{\times}$	4
30 $\boxed{\text{COS}}$	0.866
$\boxed{5)}$	3.4641
$\boxed{+}$ 5	5
$\boxed{\times}$	8.5781
6.3 $\boxed{=}$	54.0421

REGISTERAUSTAUSCH

Weiteren Bedienungskomfort bietet die Registerraustauschtaste $\boxed{x \leftrightarrow y}$. Die Betätigung dieser Taste tauscht die Zahlen in den beiden Arbeits-Registern x (Anzeige) und y aus.

$$\text{Beispiel: } \frac{20}{(4 + 3) \times 1.4} = 2.0408$$

Um die Verwendung von Speicher oder Klammern zu umgehen, berechnet man zunächst den Nenner.

Eingabe	Anzeige	Kommentar
4	4	
$\boxed{+}$	4	
3	3	
$\boxed{\times}$	7	
1.4	1.4	
$\boxed{\div}$	9.8	Der Nenner ist berechnet.
20	20	
$\boxed{x \leftrightarrow y}$	9.8	Damit man nicht fälschlicherweise Nenner durch Zähler teilt, tauscht man den Inhalt des x-Registers, 20 gegen den des y-Registers, 9.8 aus.
$\boxed{=}$	2.0408	Es wurde 20 durch 9.8 geteilt.

11. ANZEIGE VON KAPAZITÄTSÜBERLAUF UND FEHLER

Die Fehleranzeige, erkennbar am Aufleuchten der Marke Γ links oben in der Anzeige und beliebiger Zahlenwerte, erscheint in folgenden Fällen:

- 1) Der Betrag des Ergebnisses einer Rechnung ist größer $9.9999999 \cdot 10^{99}$ oder kleiner $1 \cdot 10^{-99}$ aber ungleich Null.
- 2) Division durch Null.
- 3) Die Zahl x in der Anzeige ist negativ und es werden die Funktionen $\boxed{\text{Ln}}$, $\boxed{\text{Log}}$ oder $\boxed{\sqrt{x}}$ abgerufen.
- 4) Die Zahl in der Anzeige ist Null. und es werden die Funktionen $\boxed{1/x}$, $\boxed{\text{Ln}}$ oder $\boxed{\text{Log}}$ abgerufen.
- 5) Die Funktion $\boxed{y^x}$ wird bei $y \leq 0$ abgerufen.
- 6) Der Betrag der Zahl in der Anzeige ist größer 1 und es werden die Funktionen $\boxed{\text{ARC SIN}}$ oder $\boxed{\text{ARC COS}}$ abgerufen.
- 7) Es werden mehr Klammern geöffnet, als der Rechner besitzt.

Die Fehleranzeige löscht man durch Druck auf die Taste $\boxed{\text{C/CE}}$. Dabei werden alle Speicher bis auf die M-Speicher gelöscht.

ANZEIGE BEI SCHWACHEN BATTERIEN

Wenn die Batterien nicht mehr genügend Strom liefern, leuchtet links in der Anzeige die Marke \perp auf.

12. RECHENGESCHWINDIGKEIT UND RECHENGENAUIGKEIT

Die Grundrechenarten, $1/x$, \sqrt{x} , x^2 , sowie sämtliche Speicher - und Klammerfunktionen werden in Bruchteilen von Sekunden ausgeführt. Die Ermittlung der übrigen Funktionen benötigt bis zu 2.8 Sekunden, Während der Rechenzeit ist die Anzeige gelöscht, eingegebene Befehle werden nicht ausgeführt.

Die meisten wissenschaftlichen Funktionen werden vom Rechner durch Reihenentwicklung oder andere Näherungsmethoden ermittelt. Um die Rechenzeit nicht unnötig groß zu machen, wird die Rechnung nach einer bestimmten Anzahl von Gliedern abgebrochen. Dadurch können sich geringe Abweichungen vom exakten Wert einstellen. Der Fehler ist aber bei fast jedem Problem vernachlässigbar. Er beträgt höchstens:

- ± 1 in der 8. Stelle bei den Funktionen $+$, $-$, x , \div , $1/x$, x^2 und \sqrt{x}
- ± 2 in der 8. Stelle bei allen übrigen Funktionen.

Etwas größere Fehler ergeben sich lediglich in der Nähe der Grenzen des Wertebereiches einer Funktion, etwa bei $\tan x$, wenn x sehr nahe bei 90° liegt. Bei der Ermittlung der Winkelfunktionen sehr großer Winkel kann man die volle Genauigkeit erhalten, wenn man vor Bedienung der Winkelfunktionstaste vom gegebenen Winkel entsprechend oft 360° abzieht. Die Sinus- und Tangenswerte extrem kleiner Winkel ermittelt man am besten mit der Beziehung $\sin \varepsilon = \tan \varepsilon = \varepsilon^\circ * \pi/180 = \varepsilon$

Bei umfangreichen Rechnungen können sich die Einzelfehler addieren.

13. ANWENDUNGSBEISPIELE

FREIER FALL

Die Geschwindigkeit v eines Körpers, der die Höhe h frei durchfallen hat, beträgt $v = \sqrt{2gh}$, wobei die Erdbeschleunigung den Wert $g = 9.81 \text{ ms}^{-2}$ besitzt. Welche Geschwindigkeit hat ein Körper nach 6 m freiem Fall?

$$v = \sqrt{2 \times 9.81 \times 6} \text{ m/s} = 10.8499 \text{ m/s}$$

Eingabe	Anzeige
2	2
<input type="button" value="×"/>	2
9.81	9.81
<input type="button" value="×"/>	19.62
6	6
<input type="button" value="="/>	117.72
<input type="button" value="√x"/>	10.8499
<input type="button" value="DP"/> <input type="button" value="2"/>	10.85

Bei technischen Problemen sollte die angegebene Stellenzahl in Einklang mit der Genauigkeit der gemessenen Werte stehen. Hier rundet man das Ergebnis auf 10.85 m/s.

KUGELVOLUMEN

Das Volumen einer Kugel vom Radius r beträgt $V = 4/3 \times r^3 \times \pi$. Wie groß ist das Volumen einer Kugel von 10 cm Durchmesser?

$$V = 4/3 \times 5^3 \times \pi = 523.5988 \text{ cm}^3$$

Eingabe	Anzeige
4	4
\div	4
3	3
\times	1.3333
(5	0
5	5
y^x	5
3	3
)5	125
\times	166.6667
π	3.1416
=	523.5988

BERECHNUNGEN AN DUNNEN LINSEN

Bei einer dünnen Linse kann man aus dem Abstand a zwischen Gegenstand und Linse sowie dem Abstand b zwischen Linse und Bild die Linsenbrennweite berechnen: $f = a \times b / (a + b)$

Man misst a = 32 cm und b = 81 cm. Wie groß ist die Linsenbrennweite?

$$f = \frac{32 \times 81}{32 + 81} \text{ cm} = 22,94 \text{ cm}$$

Eingabe	Anzeige
---------	---------

32	32
----	----

<input type="text" value="×"/>	32
--------------------------------	----

81	81
----	----

<input type="text" value="÷"/>	2592
--------------------------------	------

<input type="text" value="(5"/>	0
---------------------------------	---

32	32
----	----

<input type="text" value="+"/>	32
--------------------------------	----

81	81
----	----

<input type="text" value="5)"/>	113
---------------------------------	-----

<input "="" type="text" value="="/>	22.9381
-------------------------------------	---------

<input type="text" value="DP"/> <input type="text" value="2"/>	22.94
--	-------

Haben zwei Linsen f_1 und f_2 den Abstand e , so ist die Gesamtbrennweite $f = f_1 \times f_2 / (f_1 + f_2 - e)$. Wie groß ist die Gesamtbrennweite zweier 15 cm Linsen im Abstand 2 cm?

$$f = \frac{15^2}{2 \times 15 - 2} \text{ cm} = 8.036 \text{ cm}$$

Eingabe	Anzeige
15	15
$\boxed{F} \boxed{x^2}$	225
$\boxed{\div}$	225
$\boxed{(5}$	0
2	2
$\boxed{\times}$	2
15	15
$\boxed{-}$	30
2	2
$\boxed{5)}$	28
$\boxed{=}$	8.0357
$\boxed{DP} \boxed{2}$	8,036

FLÄCHENTRÄGHEITSMOMENT

Das Flächenträgheitsmoment eines Kreisrings in Bezug auf seine Symmetrieachse beträgt $J = \pi/64 [D_a^4 - D_i^4]$. D_a ist der Außen Durchmesser, D_i der Innendurchmesser des Rings.

Beispiel: $D_a = 7,64 \text{ cm}$; $D_i = 5,89 \text{ cm}$

Lösung: $J = 108,1625 \text{ cm}^4$

Eingabe Anzeige

π 3.1416

$\boxed{\div}$ 3,1416

64 64

$\boxed{\times}$ 0.0491

$\boxed{(5)}$ 0

$\boxed{(5)}$ 0

7,64 764

$\boxed{y^x}$ 7,64

4 4

$\boxed{5)}$ 3407.0102

$\boxed{-}$ 3407.0102

$\boxed{(5)}$ 0.0000

5,89 5.89

$\boxed{y^x}$ 5.89

4 4

$\boxed{5)}$ 1203.5418

$\boxed{5)}$ 2203.4684

$\boxed{=}$ 108.1625

STATISTIK

Bei einer Messreihe werden die Werte $x_1 = 3.2$, $x_2 = 3$, $x_3 = 4$, $x_4 = 3.8$ und $x_5 = 3.4$ ermittelt. Man berechne Mittelwert, Varianz und Standardabweichung.

$$\text{Mittelwert } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{1}{5} (3.2 + 3 + 4 + 3.8 + 3.4) = 3.48$$

Dabei bezeichnet n die Gesamtzahl der Messwerte, x_i sind die einzelnen Messwerte also x_1, x_2 usw.

Eingabe	Anzeige
3,2	3.2
<input data-bbox="145 678 199 719" type="button" value="+"/>	3.2
3	9
<input data-bbox="145 790 199 831" type="button" value="+"/>	6.2
4	4
<input data-bbox="145 902 199 943" type="button" value="+"/>	10,2
3.8	3.8
<input data-bbox="145 1014 199 1055" type="button" value="+"/>	14
3.4	3.4
<input data-bbox="145 1126 199 1167" type="button" value="÷"/>	17.4
5	5
<input data-bbox="145 1238 199 1279" type="button" value="="/>	3.48

$$\text{Varianz } \sigma^2 = \frac{\sum_{i=1}^n x_i^2 - \frac{\bar{x}^2}{n}}{n-1} = \frac{(3.2^2 + 3^2 + 4^2 + 3.8^2 + 3.4^2) - 5 \cdot 3.48^2}{5-1} = 0.172$$

$$\text{Standardabweichung } \sigma = \sqrt{0.172} = 0.4147$$

Eingabe	Anzeige
3.2	3.2
$\boxed{F} \boxed{x^2}$	10.24
$\boxed{+}$	10.24
3	3
$\boxed{F} \boxed{x^2}$	9
$\boxed{+}$	19.24
4	4
$\boxed{F} \boxed{x^2}$	16
$\boxed{+}$	35.24
3.8	3.8
$\boxed{F} \boxed{x^2}$	14.44
$\boxed{+}$	49.68
3.4	3.4
$\boxed{F} \boxed{x^2}$	11.56
$\boxed{-}$	61.24
$\boxed{(5}$	0
5 $\boxed{\times}$	5
3.48 $\boxed{F} \boxed{x^2}$	12.1104
$\boxed{5)}$	60.552
$\boxed{\div}$	0.688
$\boxed{(5}$	0
5 $\boxed{-}$	5
1 $\boxed{5)}$	4
$\boxed{=}$	0.172
$\boxed{\sqrt{x}}$	0.4147

ZINSESZINSRECHNUNG

Wird das Guthaben G_0 mit $p\%$ verzinst und werden die Zinsen k mal jährlich dem Guthaben zugeschlagen, so ist es nach Ablauf von n Jahren auf G ; angewachsen,

$$G_n = G_0 * \left(1 + \frac{p}{100 * k}\right)^{kn}$$

Beispiel:

Guthaben	$G_0 = 500$ DM
Zinssatz	$p = 6\%$ p.a.
Zinstermine	$k = 4$ p.a.
Anlagezeit	$n = 15$ Jahre

Eingabe	Anzeige	
1	1	
<input type="button" value="+"/>	1	Man beginnt mit dem zweiten Teil des Ausdrucks; damit spart man das Setzen einer Klammer.
<input type="button" value="(5"/>	0	
6	6	
<input type="button" value="÷"/>	6	
100	100	
<input type="button" value="÷"/>	0,06	
4	4	
<input type="button" value=")5"/>	0.015	
<input type="button" value="y^x"/>	1,015	Die Anweisung y^x veranlasst die Berechnung der Basis.
<input type="button" value="(5"/>	0	Der Exponent wird eingeklammert.
4	4	
<input type="button" value="×"/>	4	
15	15	
<input type="button" value=")5"/>	60	Der Exponent wurde berechnet
<input type="button" value="×"/>	2.4432	Multiplikationsfaktor für das Guthaben
500	500	
<input type="button" value="="/>	1221.6099	
<input type="button" value="DP"/> <input type="button" value="2"/>	1221.61	

Das Endguthaben beträgt 1221.61 DM

14. SPEICHERORGANISATION

Will man einen Rechner optimal nutzen, sollte man sich etwas mit der Organisation der Register befassen. Register sind die Rechenwerke und Speicher für Zahlen und Befehle.

1) **x-Register** : Das Anzeigeregister speichert eingegebene Zahlen und zeigt das Endergebnis.

2) **y-Register**: Dieses Arbeitsregister speichert die zweite eingegebene Zahl bei Rechenoperationen, die mit zwei Zahlen arbeiten (Befehle vom Typ a) al also bei den Grundrechenarten und y^x . Werden neue Zahlen eingegeben, so speichert das y-Register das Zwischenergebnis. Die durch Tastendruck eingegebenen Informationen verändern den Inhalt des x-Registers und in manchen Fällen auch den des y-Registers.

2.1) Zahleneingabe. Abruf des Speicherinhaltes und Eingabe von Pi (Befehle vom Typ b) überschreiben einfach den Wert im x-Register.

2.2) Ein Befehl vom Typ a berechnet das Zwischenergebnis und schreibt diesen Wert in die Register x und y.

2.3) Die Ergebnistaste im Anschluss an einen Befehl vom Typ b veranlasst die Berechnung des Ergebnisses und schreibt es in x- und y-Register.

Beispiel: $(3 + 15 - 2) \times 2 = 32$

Eingabe	Anzeige (x-Register)	y-Register
3	3	
+	3 \longrightarrow \blacktriangleright	3 + \downarrow
15	15	3 +
-	18 \longrightarrow \blacktriangleright	18 - \downarrow
2	2	18 -
x	16 \longrightarrow \blacktriangleright	16 x \downarrow
2	2	16 x
=	32 \longrightarrow \blacktriangleright	32

3) **M-Register:** M-Register sind rechnende Konstantenspeicher

4) **P-Register:** Klammerregister. Beim Öffnen der Klammer wird die Zahl im V-Register samt Rechenbefehl Typ a im P₁ Register gespeichert. Beim Schließen der Klammer wird die vorausgegangene Rechnung abgeschlossen, das Ergebnis steht im x-Register und der Inhalt des P₁-Registers wird ins y-Register geschoben.

Beispiel: $4 + 2 + (3 \times 5) = 21$

Eingabe	x-Register	y-Register	P ₁ -Register	P ₂ -Register
4	4		0	0
+	4 →	4 +	0	0
		↓		
2	2	4 +	0	0
+	6 →	6 +	0	0
[5	0	0	└─▶ 6+	0
			↓	
3	3	0	6+	0
			↓	
x	3 →	3 x	6+	0
		↓	↓	
5	5	3 x	┌ 6+	0
5]	15	6+ ←	0	0
=	21	21	0	0

Das P₂-Register wird für geschachtelte Klammern verwendet.

Beim Schließen der Klammer wird der Inhalt von P₂ nach P₁ geschoben.

5) **W-Register:** Interne Register, berechnen die Funktionen vom Typ c: $1/x$, \sqrt{x} ; x^2 , e^x , 10^x , x^2 , \ln , \log , die trigonometrischen Funktionen und deren Umkehrfunktionen. Diese Befehle entnehmen den Inhalt des x-Registers, berechnen die betreffende Funktion und schieben den Funktionswert ins x-Register zurück. Der Inhalt des y-Registers bleibt unverändert.

Beispiel: $3 + \sqrt{\log 10000} = 5$

Eingabe	x-Register	y-Register
3	3	0
+	3 \longrightarrow	3 +
		↓
10000	10000	3 +
		↓
Log	4	3 +
		↓
\sqrt{x}	2	3 +
=	5 \longrightarrow	5

privileg[®]

PR56D-NC

PROGRAMMIERANLEITUNG



Bestell-Nr. 463.333 5

1. PROGRAMM BEFEHLE

Die Programmbefehle **BST**, **SST**, **R/S**, **GOTO** und **SKP(-)** haben in den drei Stellungen "LOAD", "CLR" und "RUN" des Schiebeschalters jeweils unterschiedliche Bedeutung,

STELLUNG "RUN" DES SCHIEBESCHALTERS

Die Schalterstellung "RUN" dient zum umprogrammierten Rechnen und zur Ausführung von Programmen.

Hier bedeuten:

BST	(Back Step, Rückwärtsschritt) : Keine Funktion
SST	(Single Step, Vorwärtsschritt): Schrittweise Ausführung des Programms
R/S	(Start/Stop) : Startbefehl zur automatischen Programm-ausführung.
GOTO	(Springe nach): Anwahl eines Programmschritts. GOTO muss immer von der Nummer des angewählten Speicherplatzes (ab, zwei Ziffern von 00 bis 72) gefolgt werden.
SKP(-)	Befehl zum Überspringen des Programmteils vom momentanen Programmschritt bis zum nächsten programmierten Stop R/S

STELLUNG "LOAD" DES SCHIEBESCHALTERS

Diese Schalterstellung dient zur Eingabe eines Programms. Hier bedeuten:

BST	Rückwärtslaufen im Programm ohne zu löschen oder zu überschreiben um eine bestimmte Programmstelle aufzusuchen.
SST	Wie BST nur werden Vorwärtsschritte gewählt.
R/S	Stop befehl. An dieser Stelle wird die automatische Programmausführung in Stellung "RUN" unterbrochen.
GOTO	Befehl zum Sprung zum Speicherplatz ab.
SKP(-)	Bedingter Sprung. Der weitere Verlauf der Programmausführung hängt vom Vorzeichen in der Anzeige ab.

STELLUNG "CLR" DES SCHIEBESCHALTERS

Diese Schaltersteilung dient zum Löschen.

Hier bedeuten:

- | |
|-----|
| BST |
|-----|

 Schrittweises Rückwärtslöschen des Programmspeichers.
- | |
|-----|
| SST |
|-----|

 Schrittweises Vorwärtslöschen des Programmspeichers.
- | |
|-----|
| R/S |
|-----|

 Gesamtlöschung des Programmspeichers.
- | |
|------|
| GOTO |
|------|

 Keine Funktion.
- | |
|--------|
| SKP(-) |
|--------|

 Keine Funktion.

2. EINGABE UND ABRUF EINES PROGRAMMS

In der Stellung "LOAD" des Schiebeschalters speichert der Rechner die Reihenfolge der gedrückten Tasten. In Stellung "RUN" kann dann der automatische Ablauf dieser Tastenfolge beliebig oft abgerufen und mit verschiedenen Zahlenwerten durchgearbeitet werden. Das Programm bleibt solange gespeichert, bis man den Rechner abschaltet oder den Programmspeicher löscht.

Die Programmeingabe führt man nach folgendem Schema aus:

1) Schiebeschalter in Stellung "RUN". Man setzt den Schrittzähler des Programmspeichers mit dem Befehl **GOTO** **0** **0** auf den Platz 00. Ist bereits ein Programm gespeichert, wählt man den ersten freien Speicherplatz "ab" mit **GOTO** **a** **b** an.

2) Der Rechner wird in Stellung "LOAD". "Einlesen eines Programms" geschaltet.

3) Das zu programmierende Rechenproblem wird so eingetastet, wie man es in "RUN" Stellung durchrechnen würde. Lediglich bei Zahlenwerten muss man dem Rechner mitteilen, ob es sich um eine bei jedem Durchlauf gleichbleibende Konstante handelt, oder um eine Variable.

Konstanten bestehend aus den Zifferntasten **0** ... **9**, **•**, **+/-** oder **π** werden so eingetastet, wie sie im Rechenablauf folgen.

Variable sind Zahlenwerte, die bei jedem Programmdurchlauf einen anderen, vom Programmierer festzulegenden Zahlenwert annehmen sollen. Will man daher Variable eingeben, so muss man den automatischen Programmablauf unterbrechen. Das geschieht mit Hilfe der Taste **R/S**.

4) Zum Ablesen der Ergebnisse muss man den automatischen Programmablauf ebenfalls mit **R/S** unterbrechen.

5) Da das Programm mehrmals abgerufen werden soll, muss man den Rechner bei Programmende anweisen, wieder zum Programmanfang zurückzukehren. Das geschieht mit dem Befehl **GOTO** **a** **b**, wobei "ab" der Speicherplatz des Programmanfangs oder Schleifen anfangs ist. Hat man das Programm, wie meist üblich, beim Platz 00 begonnen, lauten die letzten 3 Befehlsschritte **GOTO** **0** **0**

Beispiel 1: Der Umfang $U = 2 r \pi$ von Kreisen mit verschiedenen Radien r soll ermittelt werden. In diesem Problem treten die Faktoren 2 und π bei jedem Durchlauf auf; sie sind also Konstanten. Der Radius ist dagegen bei jedem Durchlauf ein anderer; r ist damit eine Variable. Bei umprogrammierten Rechnen würde der Rechenablauf für das Beispiel $r = 5$ folgendermaßen aussehen:

Eingabe	Anzeige	Kommentar
2	2	Konstante
<input type="text" value="×"/>	2	
5	5	Variable r=5
<input type="text" value="×"/>	10	
<input type="text" value="π"/>	3.1416	Konstante
<input type="text" value="="/>	31.4159	Umfang

Wünscht man den Umfang von 10 verschiedenen Kreisen zu berechnen, so muss man 10 mal die Tastenfolge 2 x r x π drücken, Das sind 60 Arbeitsgänge. Dieser Aufwand wird durch das Programmieren erheblich vermindert, da alle gleichbleibenden Tastenbedienungen gespeichert werden können und automatisch ablaufen.

Programmieren des Problems:

Stellung "RUN":

Stellung" LOAD":

Schritt	Eingabe	Kommentar
00	2	
01	<input type="text" value="×"/>	
02	<input type="text" value="R/S"/>	Unterbrechung des automatischen Programmablaufs zur Eingabe der Variablen r.
03	<input type="text" value="×"/>	
04	<input type="text" value="π"/>	
05	<input type="text" value="="/>	Ermittlung des Ergebnisses
06	<input type="text" value="R/S"/>	Unterbrechung des automatischen Programmablaufs zum Ablesen des Ergebnisses.
07	<input type="text" value="GOTO"/>	
08	<input type="text" value="0"/>	
09	<input type="text" value="0"/>	Rücksprung zum Programmanfang

Die zweite Programmunterbrechung ist zum Ablesen des Ergebnisses notwendig. Ohne diesen Stop erscheint das Ergebnis nur für Bruchteile von Sekunden in der Anzeige. das Programm springt zum Schritt 00, läuft weiter bis 2 x und bleibt mit 2 in der Anzeige stehen.

Während des Programmierens erscheinen in der Anzeige 4 bis 5 Ziffern. Die rechten beiden Ziffern zeigen den Speicherplatz der folgenden Eingabe: die linken zwei bis drei Ziffern sind eine Codenummer für die Operation, die in Programmspeicher als nächster Schritt steht. Diese Ziffern sind beim erstmaligen Einlesen des Programms ohne Bedeutung. Sie sind aber eine große Hilfe beim Testen, Ändern und Korrigieren von Programmen.

Automatischer Programmablauf

Beim automatischen Programmablauf arbeitet man nach folgendem Schema:

- 1) Schiebeschaltet in Stellung "RUN"! Löschen der im Programm verwendeten Speicher. Diese Maßnahme ist nicht bei jedem Programm notwendig aber empfehlenswert.
- 2) Anwahl des Programmanfangs "ab" mit **GOTO** **a** **b**.
- 3) Starten des Programms mit **R/S**, eventuell nach vorheriger Eingabe einer Variablen.

Bei der Umfangsberechnung sieht das folgendermaßen aus:

Um den automatischen Programmablauf zu starten drückt man **GOTO** **0** **0** **R/S**. Jetzt werden alle Programmschritte bis zur nächsten Stopmarke des Programms ausgeführt. In unserem Beispiel berechnet die Programmautomatik $2x$ und bleibt stehen, Der Rechner wartet auf die Eingabe des variablen Radius r . Man tastet z.B. $r_1=5$ ein. Anschließend wird der Programmablauf mit **R/S** fortgeführt, der Rechner ermittelt $U_1=31.4159$ und bleibt beim programmierten Stop zum Ablesen des Ergebnisses stehen. Nun wird der Programmablauf von neuem gestartet, ein neuer Radius, etwa $r_2=6$ eingegeben und wieder mit **R/S** fortgeführt. Der Rechner ermittelt automatisch den Umfang $U_2=37.6991$ und ist für einen neuen Start bereit.

Eingabe	Anzeige	Kommentar
GOTO <input type="text" value="0"/> <input type="text" value="0"/>		Der erste Programmschritt 00 wird angewählt.
R/S	2	Das Programm wird gestartet, bleibt beim programmierten Stop mit 2 in der Anzeige stehen und wartet auf die Eingabe des Wertes von r.
5	5	Eingabe der Variablen r ₁
R/S	31.4159	Umfang U ₁
R/S	2	Das Programm wird erneut gestartet
6	6	Eingabe der Variablen r ₂
R/S	37.6991	Umfang U ₂

An diesem Programm stört noch, dass pro Durchlauf zweimal gestartet werden muss. Das vermeidet man, wenn man die Variable direkt anschließend an den programmierten Stop-befehl zum Ablesen des Ergebnisses stellt. Das Problem wird dazu umgeschrieben:

$$r_1 \times 2 \times \pi = U_1, r_2 \times 2 \times \pi = U_2$$

Stellung "RUN" **GOTO**

Stellung "LOAD":

Schritt	Eingabe	Kommentar
00	<input type="text" value="×"/>	Vor Start des Programmablaufs muss in der Anzeige r stehen, das dann mit 2 multipliziert wird.
01	2	
02	<input type="text" value="×"/>	
03	<input type="text" value="π"/>	
04	<input type="text" value="="/>	Ergebnis
05	R/S	Stop zum Ablesen des Ergebnisses und gleichzeitig zur Eingabe des nächsten Radius.
06	GOTO	
07	<input type="text" value="0"/>	
08	<input type="text" value="0"/>	

Die Programmausführung in Stellung "RUN" lautet nun:

Eingabe	Anzeige	Kommentar
GOTO 0 0		
5	5	r ₁
R/S	31.4159	U ₁
6	6	r ₂
R/S	376991	U ₂

Für die Berechnung eines Umfangs werden also an Stelle von 6 nurmehr 2 Eingaben benötigt. Bei der Berechnung von 10 Umfangswerten spart man 40 Eingaben. Eine enorme Zeitersparnis, selbst bei einem so einfachen Problem.

In einem Programm können auch mehrere Variable vorkommen. Die Zahl der programmierten Stops muss dann mindestens gleich der Anzahl der Variablen sein.

Beispiel 2: In einem rechtwinkligen Dreieck mit den Katheten a und b beträgt die Länge l der Hypotenuse $c = \sqrt{a^2 + b^2}$ (Satz von Pythagoras). a und b können beliebige Werte annehmen, sind also zwei Variable. Auch hier schreibt man eine Variable direkt im Anschluss an das Ergebnis.

Kommentar

"RUN": **GOTO** 0 0
 "LOAD"

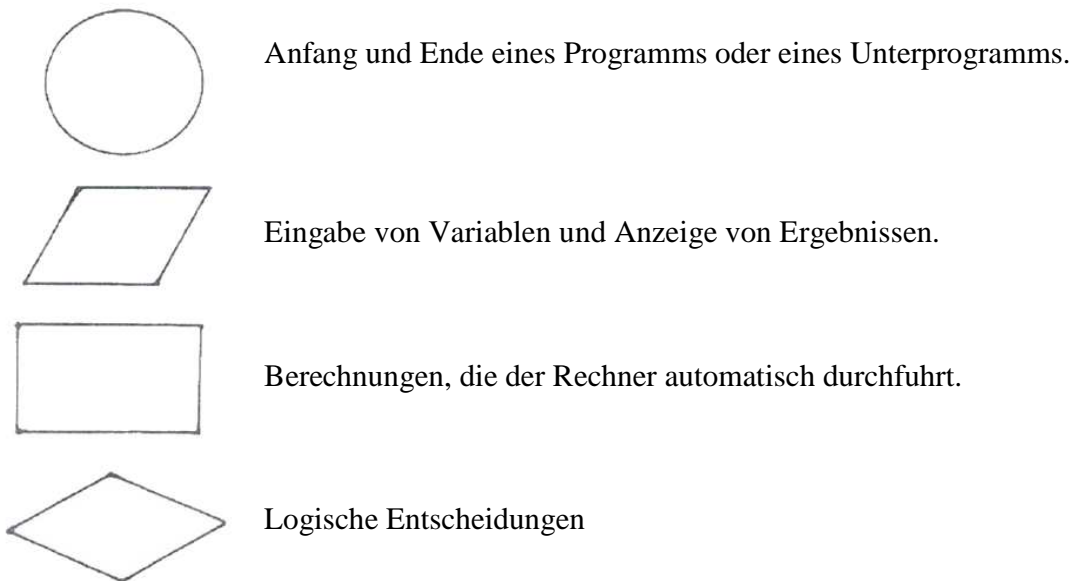
Schritt	Eingabe	Kommentar
00	F x ²	a ² . Die Bedienung der Taste F zählt nicht als eigener Programmschritt.
01	+	
02	R/S	Stop zur Eingabe von b
03	F x ²	b ²
04	=	a ² + b ²
05	√x	c
06	R/S	Stop zum Ablesen des Ergebnisses und Eingabe des nächsten a.
07	GOTO	
08	0	
09	0	Rücksprung zum Programmanfang

In der Anzeige steht jetzt rechts die 10, die nächste Eingabe war e also der 10. Programmschritt.

Programmablauf in Stellung "RUN" mit den Variablenwerten $a_1=3$, $b_1=4$ und $a_2=7$, $b_2=11$:

Eingabe	Anzeige	Kommentar
GOTO 0 0		
3	3	$a_1 = 3$
R/S	9	Der Rechner hat $a_1^2 = 9$ ermittelt und wartet auf die Eingabe von b_1
4	4	b_1
R/S	5	$c_1 = 5$
7		$a_2 = 7$. Variableneingabe im Anschluss an das Ergebnis des vorherigen Durchlaufs.
R/S	49	
11	11	b_2
R/S	13.0384	c_2

Bei komplizierten Programmen zeichnet man ein Flussdiagramm um den Ablauf des Programms besser verfolgen zu können, In Flussdiagrammen verwendet man folgende Symbole:



Die Funktion des Flussdiagramms veranschaulicht folgendes Beispiel:

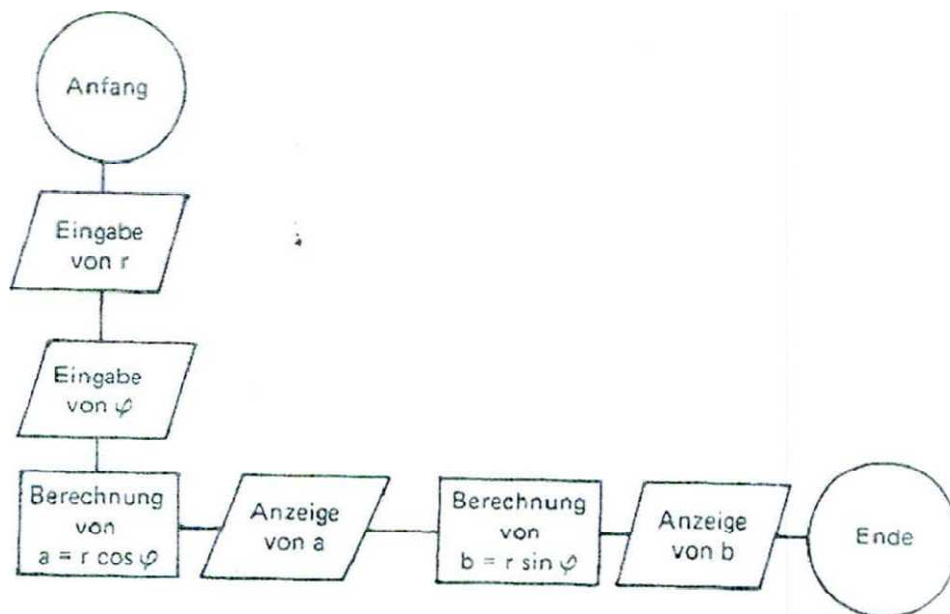
Beispiel 3:

UMWANDLUNG VON POLARKOORDINATEN IN RECHTWINKLIGE KOORDINATEN

Gegeben: Länge r und Winkel eines Zeigers.

Gesucht: Projektion des Zeigers auf die x-Achse (a) und die y-Achse (b).

Formeln: $a = r \cos \varphi$, $b = r \sin \varphi$



Das Problem wird entsprechend dem Flussdiagramm programmiert :

Stellung "RUN": **GOTO** **0** **0**

Stellung "LOAD":

Schritt	Eingabe	Kommentar
00	STO n	Speichern von r zur Berechnung von a und b
01	1	
02	R/S	Stop für die Eingabe der Variablen φ .
03	STO n	Speichern von φ
04	2	
05	COS	$\cos \varphi$
06	×	
07	RCL n	
08	1	r
09	=	$a = r \cos \varphi$ wird berechnet.
10	R/S	Stop zum Ablesen von a
11	RCL n	
12	2	
13	SIN	$\sin \varphi$
14	×	
15	RCL n	
16	1	
17	=	$b = r \sin \varphi$ wird berechnet.
18	R/S	Stop zum Ablesen von b und zum Eingeben von r.
19	GOTO	
20	0	
21	0	

An diesem Beispiel wird klar, wie einfach das Programmieren mit dem "privileg PR 56D-NC" ist: In den normalen Rechenablauf müssen lediglich Stop-marken für die Variableneingabe und zum Ablesen der Ergebnisse eingefügt werden. Am Programmende folgt der Befehl für den Rücksprung zum Programmstart.

Genauso problemlos ruft man das Programm ab. Es sollen folgende Polarkoordinaten in rechtwinklige Koordinaten umgewandelt werden:

$$r_1 = 8; \quad \varphi_1 = 12^\circ$$

$$r_2 = 10; \quad \varphi_2 = 135^\circ$$

Stellung des Winkelwahlschalters: "D"
Stellung "RUN"

Eingabe	Anzeige	Kommentar
GOTO 00		
8 R/S	8	r ₁
12	12	φ ₁
R/S	7.8252	a ₁
R/S	1.6633	b ₁
10 R/S	10	r ₁
135	135	r ₂
R/S	-7.0711	a ₂
R/S	7.0711	b ₂

3. SPEICHERN MEHRERER PROGRAMME

Sollen im Rechner gleichzeitig mehrere Programme gespeichert werden, so schreibt man die folgenden Programme jeweils direkt ans Ende des vorhergehenden. Es können beliebig viele Programme gespeichert werden. Man muss jedoch darauf achten, dass die Speicherkapazität von 72 Plätzen nicht überschritten wird.

Beispiel: Es soll im Anschluss das im letzten Kapitel besprochene Programm "Umwandlung von Polarkoordinaten in rechtwinklige Koordinaten" das Programm "Umwandlung von rechtwinkligen Koordinaten in Polarkoordinaten" geschrieben werden.

Gegeben: a, b Gesucht: r, φ

Formeln $r = \sqrt{a^2 + b^2}$; $\varphi = \arctan(b/a)$

Das vorausgehende Programm belegt die Speicherplätze von 00 bis einschließlich 21. Man beginnt also das neue Programm mit Platz 22 (Da noch genug freie Speicherkapazität ist, könnte man auch mit einem höheren Speicherplatz anfangen)

Schalterstellung "RUN": **GOTO** 22

Schalterstellung "LOAD": Der angewählte Speicherplatz. 22, steht in der Anzeige!

Schritt	Eingabe	Kommentar
22	STO _n	
23	3	
24	R/S	
25	STO _n	
26	4	
27	F x²	
28	+	
29	RCL _n	
30	3	
31	F x²	
32	+	
33	√x	
34	R/S	Stop zum Ablesen von r
35	RCL _n	

36
 37
 38
 39
 40 b / a
 41 φ
 42 Stop zum Ablesen von φ
 43
 44
 45

Nun sollen die rechtwinklige Koordinaten a 4; I) - 3 in Polar Koordinaten umgeformt werden. Anschließend werden sie Zur Kontrolle wieder' in rechtwinklige Koordinaten zurückverwandelt.

Wahlschalter in Stellung "D"

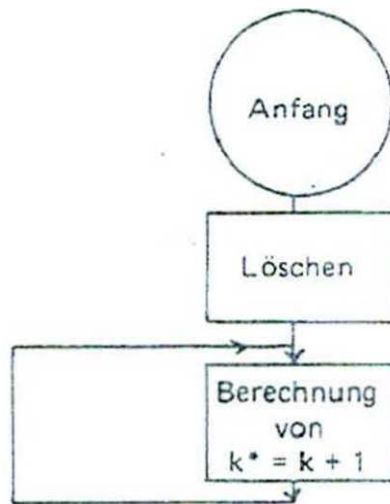
Stellung "RUN"

Eingabe	Anzeige	Kommentar
<input type="text" value="GOTO"/> <input type="text" value="2"/> <input type="text" value="2"/>		Das Programm "Umwandlung in Polar Koordinaten" steht ab Speicherplatz 22
4 <input type="text" value="R/S"/>	4	a
3	3	b
<input type="text" value="R/S"/>	5	r
<input type="text" value="STOn"/> <input type="text" value="5"/>	5	r wird für die Kontrollrechnung gespeichert.
<input type="text" value="R/S"/>	36.8699	φ
<input type="text" value="STOn"/> <input type="text" value="6"/>	36.8699	φ wird ebenfalls gespeichert
<input type="text" value="GOTO"/> <input type="text" value="0"/> <input type="text" value="0"/>	36.8699	Das Programm "Umwandlung in rechtwinklige Koordinaten" steht ab Platz 00
<input type="text" value="RCLn"/> <input type="text" value="5"/>	5	r wird abgerufen
<input type="text" value="R/S"/>	5	
<input type="text" value="RCLn"/> <input type="text" value="6"/>	36.8699	φ wird abgerufen
<input type="text" value="R/S"/>	4	a
<input type="text" value="R/S"/>	3	b

4. PROGRAMMSCHLEIFEN

Ein **GOTO** **a** **b** am Programmende weist den Rechner an, automatisch nach Programmdurchlauf zum Schritt "ab" zurückzukehren. Man kann also die gleichen Programmteile mehrmals durchlaufen, Schleifen bilden. Die Anzahl der Durchläufe einer Schleife wird mit n bezeichnet.

Beispiel 1: Man programmiere den Rechner so, dass beim Druck auf die Taste **R/S** die nächste natürliche Zahl, also 1, 2, 3, ..., n in der Anzeige erscheint (Zahlprogramm).



Stellung "RUN" **GOTO** **0** **0**
 Stellung "LOAD":

Schritt	Eingabe	Kommentar
00	C/CE	} x- und y-Register werden gelöscht, damit das Zählen mit 1 beginnt.
01	C/CE	
02	+	} Zur Zahl in der Anzeige wird 1 addiert.
03	1	
04	=	
05	R/S	
06	GOTO	} Sprung zum Schleifenanfang.
07	0	
08	3	

Da der erste Schritt bei Schleifenbeginn +1 ist, wird die Schrittweite 1 zur Zahl in der Anzeige addiert. Am Ende der Programmschleife steht $k^*=k+1$ in der Anzeige. Der Rechner springt zum Schleifenanfang 02 zurück, eine weitere 1 wird addiert und so fort.

Abruf des Programms in Stellung "RUN":

Eingabe	Anzeige	Kommentar
GOTO 0 0		
R/S	1	Die Anzeige wurde gelöscht und 1 zur 0 in der Anzeige addiert.
R/S	2	
R/S	3	
.	.	
.	.	
.	.	

Als Schrittweite kann man auch eine andere Zahl wählen. Man erhält so ein Programm für das 1 x 1, Nachteilig wäre bei diesem Programm allerdings, dass nur ein bestimmtes 1 x 1 abgerufen werden kann. Das lässt sich ändern, wenn man das gesuchte 1 x 1 in einen Speicher einliest.

Beispiel 2: Programm für das 1 x 1

Stellung "RUN":

Stellung "LOAD":

Schritt	Eingabe	Kommentar
00	<input type="text" value="C/CE"/>	
01	<input type="text" value="C/CE"/>	
02	<input type="text" value="+"/>	
03	<input type="text" value="RCLn"/>	Im Speicher 0 steht das gewünschte 1 x 1
04	<input type="text" value="0"/>	
05	<input "="" type="text" value="="/>	
06	<input type="text" value="R/S"/>	
07	<input type="text" value="GOTO"/>	
08	<input type="text" value="0"/>	
09	<input type="text" value="0"/>	

Es soll das 1 x 1 mit 8 abgerufen werden.

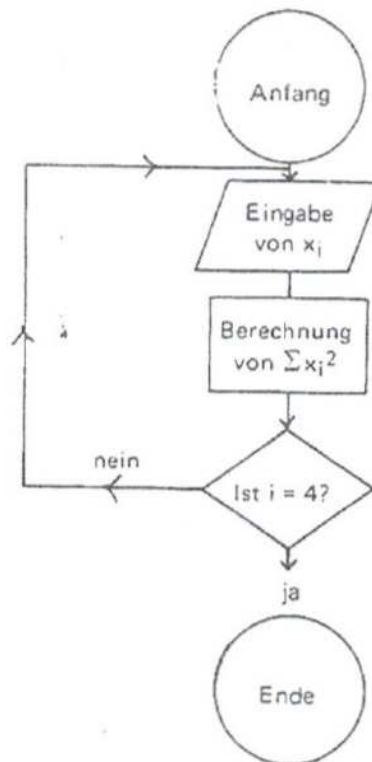
Stellung "RUN"

Eingabe	Anzeige	Kommentar
8 <input type="text" value="STOn"/> <input type="text" value="0"/>	8	Das gewünschte 1 x 1 wird n den Speicher 0 geschrieben.
<input type="text" value="GOTO"/> <input type="text" value="0"/> <input type="text" value="0"/>	8	
<input type="text" value="R/S"/>	8	
<input type="text" value="R/S"/>	16	
<input type="text" value="R/S"/>	24	
.	.	
.	.	
.	.	

Beispiel 2: Bei mehrmaliger Messung einer Größe x wurden die Werte $x_1 = 7.2$; $x_2 = 7.3$; $x_3 = 6.9$; $x_4 = 7.5$ abgelesen. Man bestimme die für die statistische Auswertung wichtige Größe

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 = \sum_{i=1}^4 x_i^2$$

Programmieren:



Stellung "RUN"; GOTO 0 0
 Stellung "LOAD":

Schritt	Eingabe	Kommentar
00	F x^2	Der Wert in der Anzeige wird Quadriert.
01	=	Neue Quadratsumme
02	+	Befehl zum Addieren des nächsten Quadrats.
03	R/S	Stop-befehl zur Eingabe eines neuen Wertes von x
04	GOTO	
05	0	
06	0	

Programmablauf in Stellung "RUN"

Eingabe	Anzeige	Kommentar
GOTO 0 0		
C/CE C/CE	0	Löschen von x- und y-Speicher, damit nicht bereits dort stehende Werte mit addiert werden.
7.2 R/S	51.84	x_1^2
7.3 R/S	105.13	$x_1^2+x_2^2$
6.9 R/S	152.74	$x_1^2+x_2^2+x_3^2$
7.5 R/S	208.99	$x_1^2+x_2^2+x_3^2+x_4^2$ Endergebnis

Beispiel 3:

Ein Satellit bewegt sich auf einer Kreisbahn um die Erde. Beginnend bei der Ausgangshöhe h_0 über der Erdoberfläche soll man in Abständen Δh Geschwindigkeit v und Umlaufdauer T berechnen.

Gegeben sind der mittlere Erdradius $R = 6370 \text{ km}$ und die Erdbeschleunigung an der Erdoberfläche $g_R = 9.81 \text{ m s}^{-2}$.

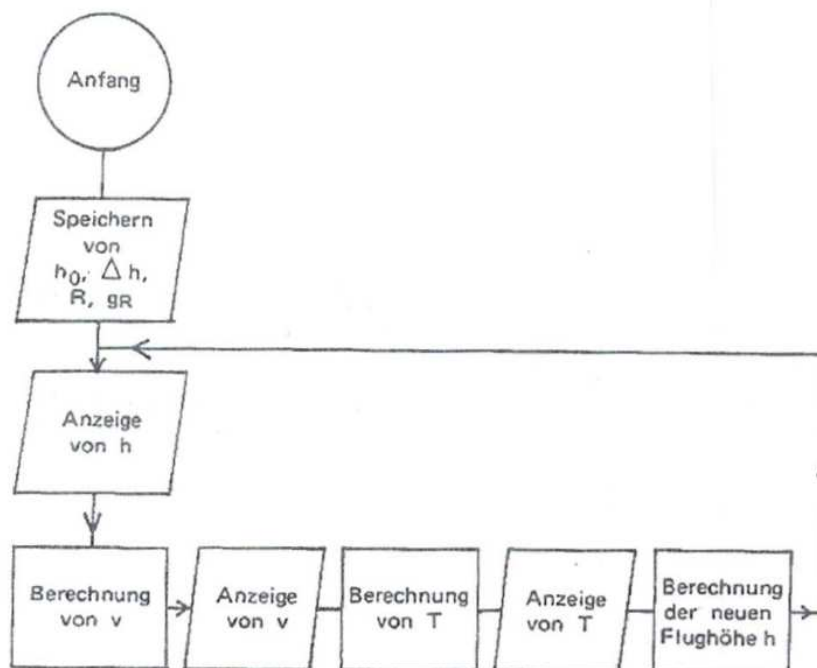
Es gilt: $r = R + h$ $h = \text{Höhe über der Erdoberfläche} = h_0 + n \Delta h$
 $v = R \sqrt{g_R/r}$
 $T = 2\pi r/v$

Es soll mit folgenden Einheiten gerechnet werden:

- Strecken in km
- Umlaufdauer in Stunden
- Geschwindigkeit in km/s

Man speichert zunächst alle Konstanten, berechnet dann mit Hilfe der gegebenen Formeln Geschwindigkeit und Umlaufdauer bei der Höhe h_0 und ermittelt am Ende des Programms mit Hilfe der Schrittweite Δh die neue Höhe h . Dann springt man an die Stelle des Programms, bei der die Geschwindigkeit berechnet wird, zurück.

Stellung "RUN"
Stellung "LOAD"



Schritt	Eingabe	Kommentar
00	<input type="text" value="STOn"/>	} Die variable Ausgangshöhe h_0 wird in M 0 gespeichert.
01	<input type="text" value="0"/>	
02	<input type="text" value="R/S"/>	Stop für die Eingabe von Δh
03	<input type="text" value="STOn"/>	} Die variable Schrittweite Δh wird in M 1 gespeichert.
04	<input type="text" value="1"/>	
05	6	
06	3	
07	7	} Der mittlere Erdradius $R = 6370$ km wird in M 2 gespeichert.
08	0	
09	<input type="text" value="STOn"/>	
10	<input type="text" value="2"/>	
11	9	
12	<input type="text" value="•"/>	
13	8	
14	1	} Die Erdbeschleunigung an der Erdoberfläche $g_R = 9.81 \cdot 10^{-3} \text{ km s}^{-2}$ wird in M 3 gespeichert.
15	<input type="text" value="EE"/>	
16	<input type="text" value="+/-"/>	
17	3	
18	<input type="text" value="STOn"/>	
19	<input type="text" value="3"/>	
20	<input type="text" value="RCLn"/>	} Beginn der Berechnung von v durch Abruf von h .
21	<input type="text" value="0"/>	
22	<input type="text" value="R/S"/>	Stop zum Ablesen der Höhe h , in der v und T berechnet wird.
23	<input type="text" value="+"/>	
24	<input type="text" value="RCLn"/>	R
25	<input type="text" value="2"/>	
26	<input type="text" value="="/>	$r = h + R$ Einheit km
27	<input type="text" value="STOn"/>	r wird zur weiteren Verwendung in M 4 gespeichert.

Schritt	Eingabe	Kommentar
28	4	
29	RCLn	
30	3	g_R
31	\div	
32	RCLn	
33	4	
34	=	g_R/r
35	\sqrt{x}	
36	\times	
37	RCLn	
38	2	R
39	=	$v=R\sqrt{g_R/r}$
40	STOn	v wird zur weiteren Verwendung in M 5 gespeichert.
41	5	
42	R/S	Stop zum Ablesen von v
43	2	Beginn der Berechnung der Umlaufdauer T
44	\times	
45	π	
46	\times	
47	RCLn	
48	4	
49	\div	
50	RCLn	
51	5	
52	\div	

Schritt	Eingabe	Kommentar
53	3	Das Ergebnis soll in Stunden angegeben werden. Umrechnungsfaktor von Sekunden in Stunden: 1/3600.
54	6	
55	0	
56	0	
57	=	Umlaufdauer T in Stunden
58	R/S	Stop zum Ablesen von T
59	RCLn	Beginn der Berechnung der neuen Höhe h
60	1	Δh
61	F Mn +	
62	0	Die neue Höhe ist $h = h_0 + \Delta h$.
63	GOTO	Rücksprung zur Berechnung von v bei der neuen Höhe $h = h_0 + \Delta h$
64	2	
65	0	

Als Zahlenbeispiel sollen Geschwindigkeit v und Umlaufzeit T ab $h_0 = 100$ km über der Erdoberfläche in schritten von 50 km bestimmt

werden: Stellung "RUN"

Eingabe	Anzeige	Kommentar
GOTO 0 0		
DP 4		
100	100	Ausgangshöhe $h_0 = 100$ km
R/S	100	
50	50	Schrittweite $\Delta h = 50$ km
R/S	100	Anzeige der Höhe h , für die v und T berechnet wird.
R/S	78437	Die Geschwindigkeit in 100 km Höhe über der Erdoberfläche beträgt 7.8437 km/s
R/S	1.4397	Die Umlaufdauer in 100 km Höhe über der Erdoberfläche beträgt 1.4397 Stunden
R/S	150	Neues $h = 100$ km + 50 km
R/S	78136	$v = 7.8136$ km/s in 150 km Höhe
R/S	1.4564	$T = 1.4564$ hin 150 km Höhe
R/S	200	$h = 150$ km + 50 km
R/S	7.7838	$v = 7.7838$ km/s in 200 km Höhe
R/S	1.4732	$T = 1.4732$ h in 200 km Höhe
R/S	250	$h = 200$ km + 50 km

Aus diesem Programm ersieht man folgende Merksätze.

1) Konstanten werden immer zuerst eingelesen. In unserem Beispiel müssen die Schritte 00 bis 19 nur einmal durchlaufen werden; die Schleife beginnt bei Schritt 20. Dadurch verringert man die Rechenzeit.

2) Fehlt es an Programmspeicher, so kann man die Konstanten auch in "RUN"-Stellung in ihre Speicher einlesen.

3) Damit man beim Programmieren stets weiß, welche Größe sich in welchem Speicher befindet, stellt man eine Liste auf:

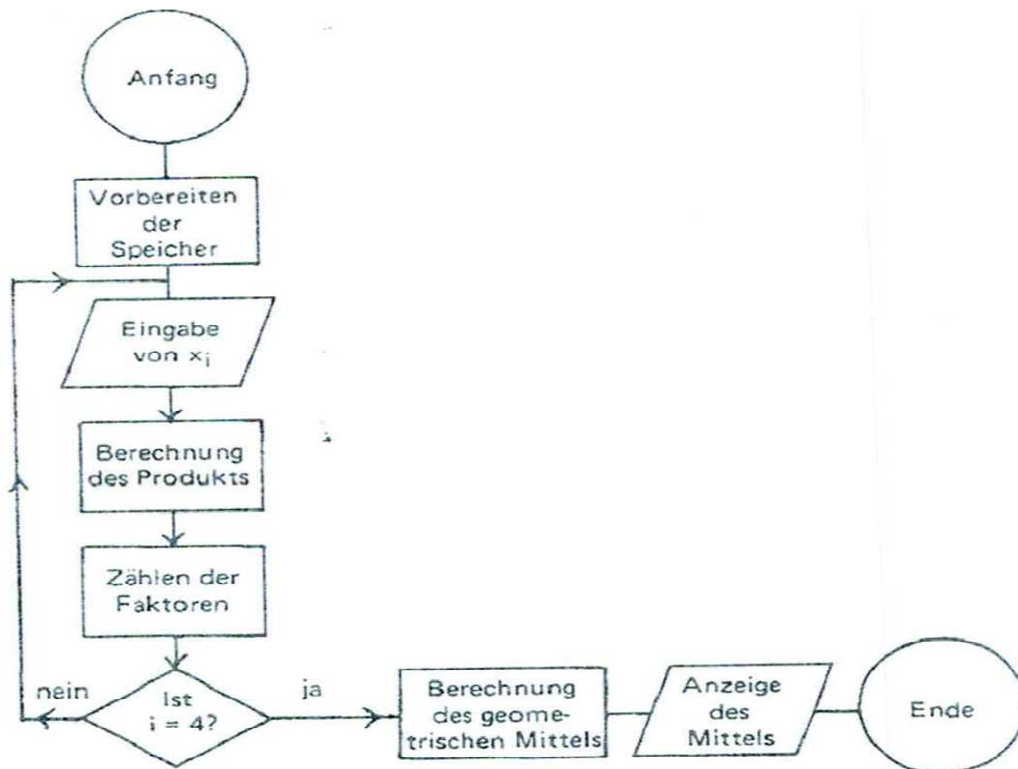
- M 0: Höhe h über der Erdoberfläche
- M 1: Schrittweite Δh
- M 2: Erdradius R
- M 3: Erdbeschleunigung an der Erdoberfläche g_R
- M 4: Abstand des Satelliten vom Erdmittelpunkt r
- M 5: Geschwindigkeit v in der Höhe h

4) Physikalische und technische Größen bestehen aus Zahlenwert und Einheit. Da jeder Rechner reine Zahlen verarbeitet, muss man immer angeben, in welchen Einheiten die Größen eingegeben und ausgegeben werden sollen.

5) Bei komplizierten Berechnungsserien lässt man den Rechner zur Orientierung bei einer übersichtlichen Bezugsgröße stoppen. In unserem Beispiel erfolgt das bei der Flughöhe h . Fehlt eine solche Größe, baut man ein Zählwerk ins Programm ein.

5. MANUELLE PROGRAMMVERZWEIGUNG

Beispiel 1: Man ermittle das geometrische Mittel $\sqrt[n]{\prod x_i}$, der Zahlen 3; 4; 7; 5, also $\sqrt[4]{3 \cdot 4 \cdot 7 \cdot 5}$.



Zunächst wird die Schleife programmiert, Dabei wird mit Hilfe der Speicher gerechnet, da das besonders übersichtlich ist. Im Anschluss an das Hauptprogramm setzt man das Unterprogramm zur Berechnung des geometrischen Mittels, das nach der Eingabe des letzten x-wertes mit **GOTO** abgerufen wird.

Stellung "RUN": GOTO 0 0
 Stellung "LOAD":

Schritt	Eingabe	Kommentar
00	C/CE	
01	C/CE	
02	STOn	} Im Speicher 1 soll die Zahl n der Eingaben stehen. Er muss daher vor der ersten Eingabe gelöscht sein.
03	1	
04	1	
05	STOn	} Im Speicher 2 soll das Produkt gebildet werden. Die erste Eingabe wird mit 1 multipliziert. Eine 0 im Speicher 2 würde diese Eingabe löschen!
06	2	
07	R/S	} Die Vorbereitung der Speicher ist abgeschlossen. Ein Stop für die Zahleneingabe wird programmiert.
08	F Mn x	
09	2	
10	1	} Produktbildung in Speicher 2
11	F Mn +	
12	1	} Zählen der Faktoren in Speicher 1
13	GOTO	
14	0	} Rücksprung zum Schleifenanfang
15	7	

Nach der nten Eingabe steht im Speicher 1 die Zahl n und im Speicher 2 das Produkt $x_1 \times x_2 \times x_3 \times \dots \times x_n$. Nun folgt das Unterprogramm zur Berechnung der nten Wurzel.

- 16 RCLn
- 17 2
- 18 y^x
- 19 RCLn
- 20 1
- 21 1/x
- 22 =
- 23 R/S

Programmablauf in Stellung "RUN":

Eingabe	Anzeige	Kommentar
GOTO 0 0		
R/S	<u>1</u>	Erster Start zur Vorbereitung der Speicher.
3 R/S	<u>1</u>	Eingabe von x_1
4 R/S	1	Eingabe von x_2
7 R/S	1	
5 R/S	1	Letzte Eingabe, $n = 4$
GOTO 1 6	1	Abruf des Unterprogramms (Platz 16)
R/S	4.5270	Geometrisches Mittel der eingegebenen Werte.

Zur Kontrolle überzeuge man sich vom Speicherinhalt:

RCLn 1	4	$n = 4$
RCLn 2	420	$3 \times 4 \times 7 \times 5 = 420$

6. RECHNERGESTEUERTE PROGRAMMVERZWEIGUNG (BEDINGTER SPRUNG)

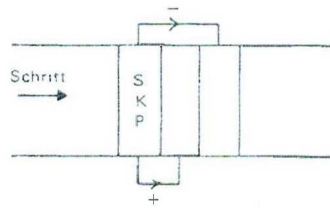
Die Taste **SKIP(-)** prüft, ob in der Anzeige ein negatives Zeichen steht.

a) In der Anzeige steht kein Vorzeichen

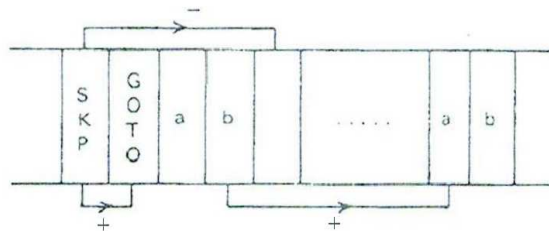
Das Programm arbeitet mit dem nächsten Schritt weiter.

b) In der Anzeige steht ein negatives Vorzeichen:

Der auf **SKIP(-)** folgende Programmschritt wird übersprungen. Alle mit **F** abgerufenen Funktionen und **GOTO** a b gelten hier als ein Schritt. Speicher und Speicherabruf dagegen belegen 2 Schritte.

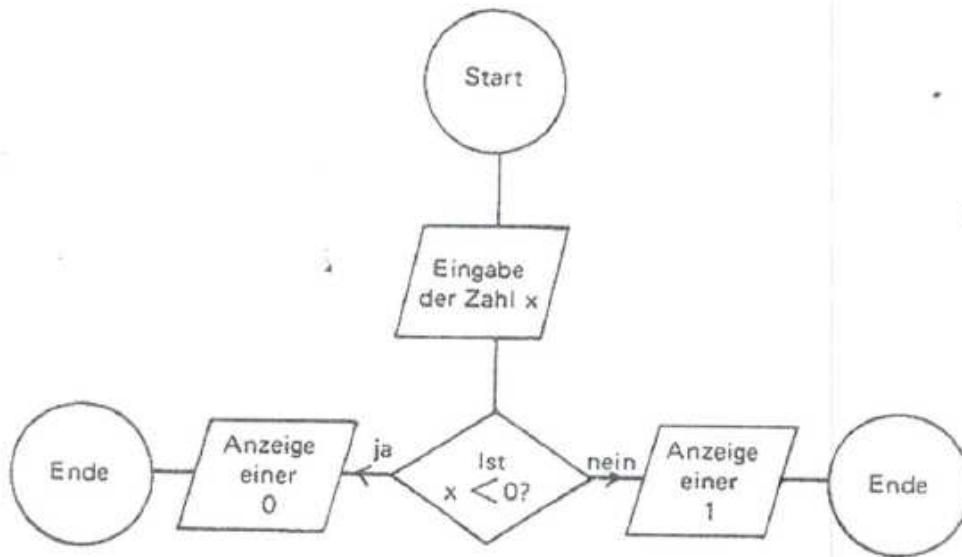


bzw



Die **SKIP(-)** ermöglicht es also, logische Entscheidungen zu treffen.

Beispiel 1: Zur Demonstration der Wirkungsweise der Taste **SKIP(-)** soll das "Signum" einer Zahl bestimmt werden: Gibt man eine beliebige positive Zahl ein, so soll der Rechner eine 1 anzeigen; bei einer negativen Zahl dagegen eine 0.



Stellung "RUN": GOTO 0 0
 Stellung "LOAD"

Schritt	Eingabe	Kommentar
00	SKP(-)	Abfrage des Vorzeichens: pos: Sprung zu Schritt 9 und Anzeige einer 1; neg: Sprung über das GOTO 0 0 zu Schritt 4 und Anzeige einer 0.
	+ / -	
01	GOTO	Loschen der Anzeige, also 0!
02	0	
03	9	
04	C/CE	
05	R/S	
06	GOTO	} Anzeige einer 1!
07	0	
08	0	
09	C/CE	
10	1	
11	R/S	
12	GOTO	
13	0	
14	0	

Programmausführung in Stellung "RUN"

Eingabe	Anzeige	Kommentar
<input type="text" value="GOTO"/> <input type="text" value="0"/> <input type="text" value="0"/>		
<input type="text" value="C/CE"/> 234	234	
<input type="text" value="R/S"/>	1	Positive Zahl
<input type="text" value="C/CE"/> 32 <input type="text" value="+/-"/>	-32	
<input type="text" value="R/S"/>	0	Negative Zahl
<input type="text" value="C/CE"/> 0	0	
<input type="text" value="R/S"/>	1	Positive Zahl
<input type="text" value="C/CE"/> 0 <input type="text" value="+/-"/>	-0	
<input type="text" value="R/S"/>	0	Negative Zahl

Da eine berechnete 0 kein Vorzeichen hat, wird auch bei 0 in der Anzeige eine 1 angegeben. Bei der Eingabe von 0 dagegen erscheint als Signum 0.

Das Programm hat noch den Schönheitsfehler, dass man vor jeder Zahleneingabe löschen muss, damit die eingegebene Zahl nicht hinter die 1 eines vorher berechneten Signums geschrieben wird. Das kann man ändern, wenn man vor jedes ein setzt. Die nach einem eingegebene Zahl überschreibt die Anzeige.

Man beachte bei diesem Einfügen eines Schritts, dass die bisherige Sprungadresse 9 jetzt in 10 umgewandelt werden muss.

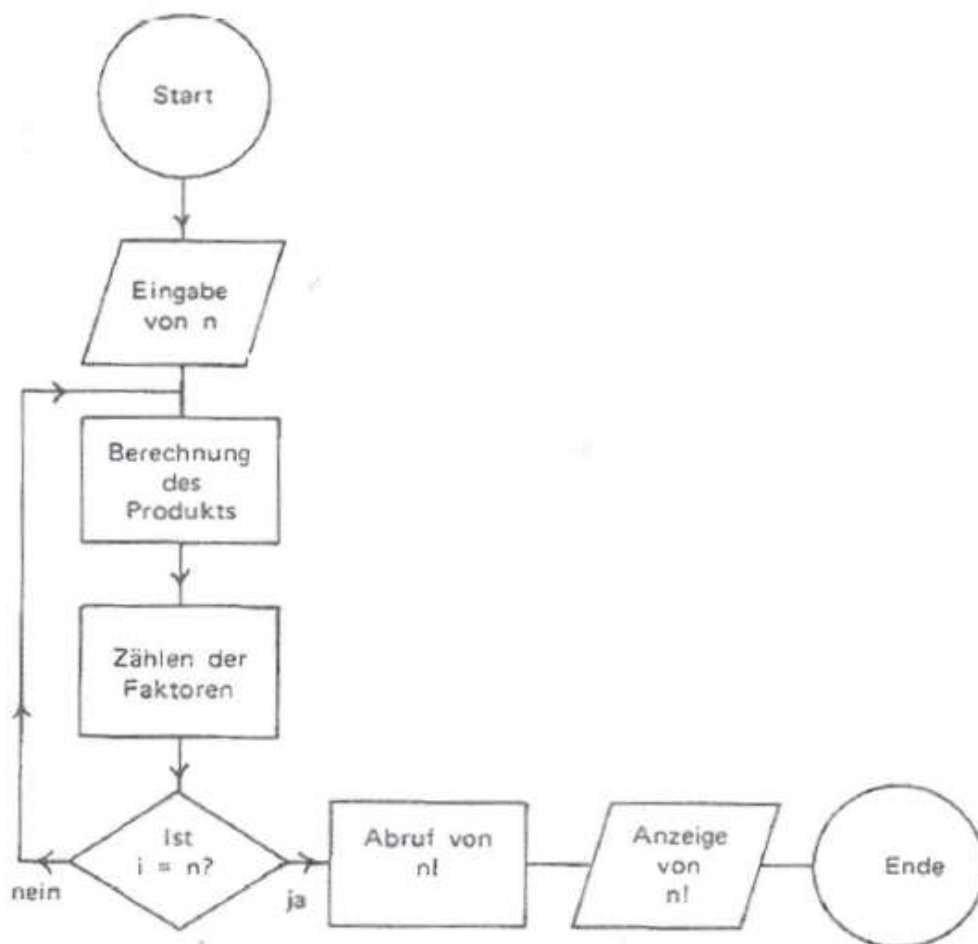
Beispiel 2: Es soll die Fakultät $n! = 1 \times 2 \times 3 \times \dots \times x_i \times \dots \times x_n$, berechnet werden. Man verwendet folgende Speicher:

- Speicher 1: Hier wird n gespeichert.
- Speicher 2: Hier wird das Produkt $x_i!$ berechnet.
- Speicher 3: Hier werden die berechneten Glieder, also i gezählt.

Die Belegung der Speicher bei den ersten 4 Durchläufen zeigt folgendes Diagramm:

	Speicher 1	Speicher 2	Speicher 3
Vorbereitung der Speicher	n	1	0
Durchlauf 1	n	1	1
Durchlauf 2	n	2	2
Durchlauf 3	n	6	3
Durchlauf 4	n	24	4

Nach jedem Durchlauf muss abgefragt werden, ob in Speicher 3 bereits n steht. Man bildet daher die Differenz $n - (\text{Inhalt von Speicher 3})$ und fragt ab, ob diese Differenz 0 ist. Da $+0$ noch als "positiver" Wert zählt, ermittelt man $n - (\text{Inhalt von Speicher 3}) - 1$. Dieser Wert wird zum ersten mal negativ, wenn der Inhalt von Speicher 3 gleich n ist. Bei positiver Differenz ist n noch nicht erreicht und man springt zum Schleifenanfang. Bei negativer Differenz programmiert man den Abruf des Speichers 2 und stoppt das Programm.



Stellung "RUN": GOTO 0 0
 Stellung "LOAD":

Schritt	Eingabe	Kommentar
00	C/CE	
01	C/CE	
02	STOn	
03	3	Zählspeicher wird gelöscht
04	1	
05	STOn	
06	2	1 wird in den Produktspeicher geschrieben
07	R/S	Einlesen von n
08	STOn	
09	1	Speichern von n
10	1	Beginn der Schleife
11	F Mn+ 1	
12	3	Zählwerk wird um 1 erhöht
13	RCLn	
14	3	
15	×	
16	RCLn	
17	2	
18	=	Berechnung des Produkts
19	STOn	
20	2	Neues Produkt im Speicher 2
21	RCLn	Beginn der Abfrage
22	1	
23	-	
24	RCLn	

25	<input type="text" value="3"/>	
26	<input type="text" value="-"/>	
27	<input type="text" value="1"/>	
28	<input type="text" value="="/>	n - i - 1
		Abfrage:
29	<input type="text" value="SKP(-)"/>	≥ 0 nochmals durchlaufen $n - i - 1 = $ $\backslash < 0$ aufhören
30	<input type="text" value="GOTO"/>	
31	<input type="text" value="1"/>	
32	<input type="text" value="0"/>	
33	<input type="text" value="RCLn"/>	
34	<input type="text" value="2"/>	
35	<input type="text" value="R/S"/>	
36	<input type="text" value="GOTO"/>	
37	<input type="text" value="0"/>	
38	<input type="text" value="0"/>	

Stellung "RUN"

Eingabe	Anzeige	Kommentar
GOTO 0 0		
R/S		Vorbereiten der Speicher
0 R/S		0!
R/S		Erneutes Vorbereiten der Speicher
1 R/S		1!
R/S		
2 R/S	2	2!
R/S		
5 R/S	120	5!
R/S		
69 R/S	1.7112 98	$69! = 1.7112 \cdot 10^{98}$
R/S		
70 R/S	Γ	Überlauf, Ergebnis größer 10^{100} !

Bei der Berechnung der Fakultät großer Zahlen müssen viele Schritte ausgeführt werden und das Programm läuft entsprechend lange.

7. ÜBERSPRINGEN VON PROGRAM TEILEN

In Stellung "RUN" bewirkt die Bedienung von **SKIP(-)**, dass der Programmteil bis zur nächsten programmierten Stop-Marke **R/S** übersprungen wird. Man führt das Programm mit Start, also **R/S** weiter; der weitere Programmablauf beginnt mit dem auf **R/S** folgenden Programmschritt.

Beispiel: Im Programmspeicher stehen die Zahlen 1; 2; 3.

Sie sollen auf die übliche Weise und mit **SKIP(-)** abgerufen werden.

Stellung "RUN": **GOTO** **0** **0**

Stellung "LOAD"

Schritt	Eingabe
00	C/CE
01	1
02	R/S
03	C/CE
04	2
05	R/S
06	C/CE
07	3
08	R/S

Stellung "RUN":

Eingabe	Anzeige	Kommentar
GOTO 0 0		
R/S	1	Üblicher Abruf mit Start, R/S
R/S	2	
R/S	3	
GOTO 0 0	3	
SKIP(-)	3	Der Programmteil Schritt 00 bis 02 wurde übersprungen.
R/S	2	Die Programmabarbeitung beginnt mit Schritt 03.
GOTO 0 0	2	
SKIP(-) SKIP(-)	2	Schritt 00 bis 05 werden übersprungen
R/S	3	Beginn mit Schritt 06

Die Taste **SKIP(-)** ist also auch wertvoll, wenn man bei der Programm Abarbeitung Teile überspringen will. Ist man sich bei mehrmaliger Bedienung von **SKIP(-)** nicht mehr sicher, an welcher Stelle des Programms man arbeitet, schaltet man um auf Schalterstellung "CLR". Der Programmschritt steht jetzt rechts oben in der Anzeige.

8. KORREKTUREN VON PROGRAMMFEHLERN UND ERGÄNZUNG EINES PROGRAMMS

Zu Korrektur und Überprüfung des Programms dienen die Schalterstellung "CLR" und die Tasten **BST** und **SST**

8.1 AUFFINDEN VON PROGRAMMFEHLERN

Ist ein Programm geschrieben, sollte man es zunächst auf eventuelle Fehler untersuchen. Dazu gibt man die Variablen ein, für die man das Ergebnis bereits kennt. Liegen solche Variablen nicht vor, wählt man Werte, die sich leicht verarbeiten lassen und berechnet das Ergebnis. Die Zahlen 0 und 1 sollte man zum Tasten nicht verwenden. Bei 1 als Testvariable würde man z.B. nicht erkennen, wenn man vergessen hat zu Quadrieren. Weicht das Ergebnis der automatischen Programmausführung vom erwarteten Wert ab, findet man den Fehler durch schritt weises Abarbeiten des Programms. Dazu schiebt man den Wahlschalter in Stellung "RUN" und arbeitet mit der Taste **SST**. Jeder Druck auf **SST** führt den nächsten Programmschritt aus. Man kann also die Abarbeitung des Programms schrittweise verfolgen und erkennen, an welcher Stelle das Programm von der Rechenvorschrift abweicht, wo man das Programm korrigieren muss.

Diese Funktion der Taste **SST** soll an dem Beispiel "Satz des Pythagoras" überprüft werden. Das Programm lautet:

Schritt	Eingabe
00	F x²
01	+
02	R/S
03	F x²
04	=
05	√x
06	R/S
07	GOTO
08	0
09	0

Es soll schrittweise mit den Variablen a = 3 und b = 4 überprüft werden.

Stellung "RUN" .

Eingabe	Anzeige	Schritt
GOTO 0 0		
3 SST	9	00
SST	9	01
SST	9	02
4 SST	16	03
SST	25	04
SST	5	05
SST	5	06
SST	5	07 bis 09

Der Rechner steht jetzt wieder bei Schritt 00, wie man sich durch Umschalten auf "CLR" überzeugen kann.

Man beachte, dass die Eingaben **F TASTE** sowie **GOTO a b** als ein Schritt zählen.

8.2 KORREKTUR VON PROGRAMMFEHLERN

8.2.1. ÜBERSCHREIBEN EINES FALSCHEN PROGRAMMSCHRITTS

Steht z.B. auf Platz 05 unseres Testprogramms anstelle von \sqrt{x} ein x^2 , so wird der Schritt 05 die Zahl 625 anstelle von 5 in die Anzeige schreiben. Diesen Fehler korrigiert man wie folgt:

Stellung "RUN": **GOTO** 0 5

Stellung "LOAD": In der Anzeige steht **18 05**, am Platz 05 steht der Befehl mit der Codezahl 18. Wie man aus der Tabelle am Ende der Bedienungsanleitung sieht, bedeutet die Codezahl 18 den Befehl x^2 . Man tastet \sqrt{x} ein. Anstelle von x^2 steht jetzt \sqrt{x} . In der Anzeige erscheint der nächste Schritt 90. 06, der Stop-Befehl **R/S** auf Platz 06. Durch weiteres Drucken von **SST** und Vergleich mit der Tabelle kann man auch auf diese Weise das Programm prüfen.

Will man mehrere Befehle überschreiben, so muss man den Speicherplatz nicht jedesmal mit **GOTO** neu anwehten, man kann vielmehr in "LOAD"-Stellung schrittweise mit **SST** zu höheren und mit **BST** zu niedrigeren Speicheradressen laufen.

8.2.2 LÖSCHEN VON PROGRAMMEN UND PROGRAMMTEILEN GELÖSCHT WIRD IN SCHALTERSTELLUNG "CLR".

Tastendruck **R/S** bewirkt totales Löschen des Programmspeichers.

Tastendruck **SST** löscht den betreffenden Schritt durch Überschreiben mit einem Leerzeichen Codezahl 99. Gleichzeitig läuft der Programmspeicher einen Schritt weiter, sodass mehrere Plätze gelöscht werden können. Das Leerzeichen wird bei der Programmausführung einfach überlesen.

Tastendruck **BST** arbeitet genauso wie **SST**, nur wird der Programmspeicher jeweils um einen Schritt zurückgesetzt.

8.2.3 EINSETZEN VON SCHRITTEN

Das Einsetzen von Schritten in ein fortlaufendes Programm ist nicht ohne weiteres möglich. Man kann aber für eventuelle Korrekturen Speicherplätze mit Leerzeichen freihalten: An beliebiger Stelle des Programms wird auf "CLR" geschaltet und **SST** entsprechend der Zahl der freizuhaltenden Plätze gedrückt. Das ist vor allem bei langen Programmen empfehlenswert. Man muss dann, wenn das Einsetzen eines Schrittes nötig wird, nicht das ganze Programm neu schreiben, sondern nur einen kleinen Teil davon, nämlich den bis zur nächsten Leerstelle.

9. KURZZUSAMMENFASSUNG

PROGRAMMIEREN

1. "LOAD" $\overline{\text{R/S}}$

Wird verwendet a) zur Eingabe von Variablen
 b) zur Anzeige des Ergebnisses

Die Programmausführung wird in der "RUN"-Stellung bei der StopMarke R/S unterbrochen und mit dem Startbefehl R/S fortgeführt.

2 "LOAD" $\overline{\text{GOTO}}$ $\overline{\text{a}}$ $\overline{\text{b}}$

Unbedingter Sprung zum Programmschritt ab.

3. "LOAD" $\overline{\text{SKP(-)}}$

Bedingter Sprung. Falls in der Anzeige ein negatives Zeichen steht, überspringt der Rechner bei der Programmausführung einen Schritt. Fehlt das Vorzeichen, fährt das Programm mit dem nächsten Schritt weiter. $\overline{\text{GOTO}}$ $\overline{\text{a}}$ $\overline{\text{b}}$ und $\overline{\text{F}}$ $\overline{\text{TASTE}}$ zählen als ein Schritt; $\overline{\text{RCLn}}$ $\overline{\text{a}}$ und $\overline{\text{STOn}}$ $\overline{\text{a}}$ als zwei Schritte.

4. SCHREIBEN MEHRERER PROGRAMME

Programme werden direkt hintereinander geschrieben,

5. EINLESEN VON LEERZEICHEN

Die Taste $\overline{\text{SST}}$ fügt in Stellung "CLR" Leerzeichen (Code 99) ins Programm ein. Diese Schritte werden überlesen. Man kann sie später durch Befehle überschreiben.

LOSCHEN

1. "CLR" $\overline{\text{R/S}}$

Totales Löschen des Programmspeichers

2. "CLR" $\overline{\text{SST}}$

Löschen des betreffenden Schritts durch Überschreiben mit einem Leerzeichen. Der Programmspeicher wird dabei um einen Schritt vorwärts gesetzt (Vorwärtlöschen).

3, "CLR" $\overline{\text{BST}}$

Wie 2., nur wird der Speicher um einen Schritt zurückgesetzt (Rückwärtlöschen) .

KORRIGIEREN

1. "LOAD" Einsetzen eines Befehls an Stelle eines Leerzeichens.
2. "LOAD" Überschreiben von Befehlen.

Der zu überschreibende Schritt wird gewählt:

- a) "RUN" **GOTO** **a** **b**
- b) "LOAD" **SST** Vorwärtslaufen bis zum betreffenden Schritt.
- c) "LOAD" **BST** Rückwärtslaufen bis zum betreffenden Schritt.

PROGRAMMABLAUF

1. "RUN" **GOTO** **a** **b**
Aufsuchen des Programmbeginns

- 2 "RUN" **R/S**
Abarbeiten des Programms vom momentanen Programmschritt bis zum nächsten programmierten Stop **R/S**.

- 3 "RUN" **SKP(-)**
Überspringen des Programmteils bis zum nächsten ~.

TESTPROZEDUREN

1. ÜBERPRÜFEN DER GESPEICHERTE PROGRAMMSCHRITTE

"LOAD" In der Anzeige erscheint links die Codezahl für den Befehl, rechts die Nummer des Programmschritts, der auf dem nächsten Platz steht. Druck auf **BST** zeigt den vorhergehenden Schritt, Druck auf **SST** den nächstfolgenden.

2. SCHRITTWEISES ABARBEITEN DES PROGRAMMS

"RUN" Jeder Druck auf **SST** führt den nächsten Programmschritt aus. **F TASTE** und **GOTO a b** zählen wieder als ein Schritt.

10. CODE-TABELLE

Befehl	Code	Befehl	Code
0	100	e^x	42
1	101	10^x	43
2	102	Mn+	50
3	103	Mn-	51
4	104	Mn×	52
5	105	Mn÷	53
6	106	STOn	55
7	107	RCLn	56
8	108	C/CE	60
9	109	+/-	62
+	10	EE	63
-	11	$x \leftrightarrow y$	64
×	12	π	65
÷	13	•	70
y^x	14	DP	71
1/x	17	=	80
x^2	18	(_s	81
\sqrt{x}	19) _s	82
sin	20	R/S	90
cos	21	GOTO	93
tan	22	SKP(-)	96
arcsin	30	Leerzeichen	99
arccos	31		
arctan	32		
Ln	40		
Log	41		