

**SHARP**®

# SCHENCOMPUTER

MODELL

# PC-1450

## BEDIENUNGSANLEITUNG



$$c = \sqrt{a^2 + b^2 - 2ab\cos\theta}$$

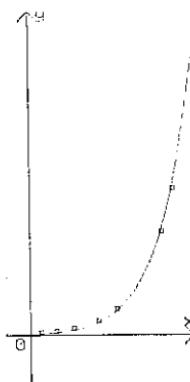


# FARBDRUCK

## ★ Exponentielle Regression und Plotten

(Siehe Seite 250)

n=9.999942306E-01  
a=4.960331516  
b=2.03657723



\* Estimation \*

x=2	y=20.45265825
x=4	y=84.3312981
x=6	y=347.7185094
x=8.5	y=495.4930476

Maßstab um 60% verkleinert

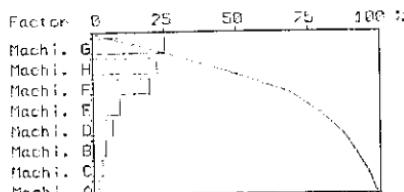
## ★ Pareto-Diagramm

(Siehe Seite 254)

Pareto

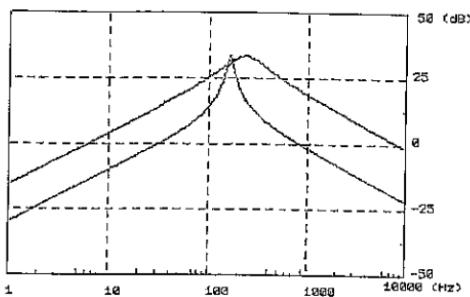
DNG date: 1985/2/15

No.	Factor	Defect	Ratio %
1	Machi. G	100	25.97
2	Machi. H	90	23.37
3	Machi. F	80	20.77
4	Machi. E	40	10.38
5	Machi. D	30	7.79
6	Machi. B	20	5.19
7	Machi. C	15	3.89
8	Machi. A	10	2.59



Maßstab um 55% verkleinert

**★ Frequenzgang-Kurve**  
(Siehe Seite 267)

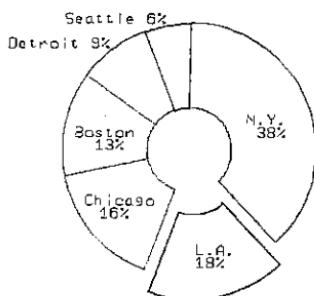


$$\begin{aligned} A(0) &= 1007520 & B(0) &= 0 \\ A(1) &= 198 & B(1) &= 5000 \\ A(2) &= 1 \end{aligned}$$

Maßstab um 85% verkleinert

**★ Kreisgrafik**  
(Siehe Seite 272)

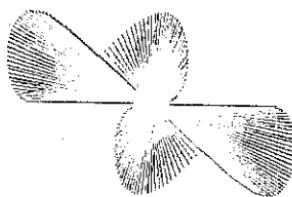
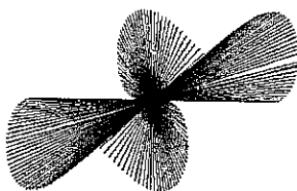
TTL Amount



Branch	Sales
N.Y.	630
L.A.	297
Chicago	270
Boston	219
Detroit	156
Seattle	98

Maßstab um 55% verkleinert

**★ 3-dimensionale Grafik**  
(Siehe Seite 283)



Maßstab um 60% verkleinert

# **SHARP PC-1450**

## **BEDIENUNGSANLEITUNG**

Liebe Leser!

Wir haben uns bemüht, dieses Handbuch ohne Fehler zu erstellen. Doch auch bei der sorgfältigsten Prüfung kann noch etwas übersehen werden. Wenn Sie daher Fehler finden und/oder den einen oder anderen Verbesserungsvorschlag zu diesem Handbuch haben, so teilen Sie uns dies bitte mit.

Vielen Dank im voraus

**SHARP ELECTRONICS  
HAMBURG**

**EINLEITUNG**

I	Allgemeines . . . . .	8
II	Betriebshinweise . . . . .	9
III	Stromversorgung . . . . .	11
IV	Tastaturabdeckung . . . . .	17
V	Richtige Behandlung des Computers . . . . .	18

**TEIL I – THEORIE**

1.1	<b>GRUNDLAGEN SHARP PC-1450</b> . . . . .	20
1.1.1	Einschalten des Computers . . . . .	20
1.1.2	Ausschalten des Computers . . . . .	20
1.1.2.1	Automatisch . . . . .	20
1.1.2.2	Manuell . . . . .	20
1.1.3	Bedienelemente . . . . .	21
1.1.4	Tastenfunktionen . . . . .	22
1.1.5	Die Anzeige . . . . .	24
1.1.6	Wahl der Betriebsart . . . . .	25
1.1.6.1	CAL-Mode . . . . .	26
1.1.6.2	RUN- und PRO-Mode . . . . .	26
1.2	<b>DIE RAM-KARTE</b> . . . . .	27
1.2.1	Installation der RAM-Karte (CE-201M, CE-202M) . . . . .	28
1.2.2	Entnahme der RAM-Karte (CE-201M, CE-202M) . . . . .	30
1.2.3	Installation der RAM-Karte (CE-211M) . . . . .	31
1.2.4	Entnahme der RAM-Karte (CE-211M) . . . . .	32
1.2.5	Verwendung der RAM-Karte mit anderen Modellen . . . . .	33
1.3	<b>GRUNDLAGEN SHARP CE-126P</b> . . . . .	35
1.3.1	Allgemeines . . . . .	35
1.3.2	Verwendung als Drucker . . . . .	36
1.3.2.1	Manueller Druckbetrieb (Protokollerstellung) . . . . .	36
1.3.2.2	Programmgesteuerter Ausdruck . . . . .	37
1.3.3	Verwendung des integrierten Kassetten-Interfaces . . . . .	38
1.3.3.1	Auswahl des Kassettenrekorders . . . . .	38
1.3.3.2	Anschluß des Kassettenrekorders an das CE-126P . . . . .	39
1.3.4	Datenspeicherung auf Magnetband . . . . .	39
1.3.4.1	Allgemeines . . . . .	39
1.3.4.2	Speichern von Daten und Programmen . . . . .	40
1.3.4.3	Überprüfen der Abspeicherung . . . . .	41

1.3.4.4	Laden von Programmen oder Daten . . . . .	41
1.4	SERIELLE E/A-FUNKTIONEN . . . . .	42

## TEIL II – PRAXIS

2.1	SHARP PC-1450 – EINSATZ ALS WISSENSCHAFTLICHER RECHNER . . . . .	46
2.1.1	Betriebshinweise . . . . .	46
2.1.1.1	Zweite Funktionsebene der Tastatur; Merkmale; Editier- und Korrekturmöglichkeiten . . . . .	46
2.1.1.2	Anzeige/Display-Format und Symbole . . . . .	53
2.1.1.3	Grundeinstellung . . . . .	54
2.1.1.4	Löschen der Eingabe oder einer Fehlermeldung . . . . .	55
2.1.1.5	Auf-/Abrundung bzw. Festlegung der Nachkommastellen . . . . .	56
2.1.1.6	Wissenschaftliche Schreibweise . . . . .	57
2.1.1.7	Umwandlung des Winkels und der Zeit . . . . .	58
2.1.1.8	Koordinatenumwandlung . . . . .	59
2.1.1.9	Umwandlung "Hexadezimal – Dezimal" und Rechnen mit Zahlen in Hexadezimal-Schreibweise . . . . .	59
2.1.2	Normale Berechnungen . . . . .	62
2.1.2.1	Addition und Subtraktion . . . . .	63
2.1.2.2	Multiplikation und Division . . . . .	63
2.1.2.3	Potenz- und Wurzelfunktion . . . . .	64
2.1.2.4	Prozentrechnung . . . . .	65
2.1.2.5	Reziprok-Rechnung . . . . .	65
2.1.2.6	Speicherrechnung . . . . .	65
2.1.2.7	Vorrangordnung und Verwendung der "Klammern"-Tasten . . . . .	66
2.1.3	Wissenschaftliche Berechnungen . . . . .	69
2.1.3.1	Trigonometrische und inverse trigonometrische Funktionen (Arcusfunktionen) . . . . .	69
2.1.3.2	Hyperbel- und inverse Hyperbelfunktionen (Arealfunktionen) . . . . .	69
2.1.3.3	Logarithmische Funktionen . . . . .	70
2.1.3.4	Exponentialfunktionen . . . . .	70
2.1.3.5	Fakultät . . . . .	70
2.1.4	Statistische Berechnungen . . . . .	71
2.1.5	Rechenbereich . . . . .	77
2.2	SHARP PC-1450 – EINSATZ ALS BASIC-RECHNER . . . . .	81
2.2.1	Rechnen ohne Programmunterstützung (RUN-Mode) . . . . .	81

2.2.1.1	Grundrechnungsarten . . . . .	81
2.2.1.2	Rechengenauigkeit . . . . .	83
2.2.1.3	Wissenschaftliche Schreibweise . . . . .	84
2.2.1.4	Editier-/Korrekturmöglichkeit . . . . .	85
2.2.1.5	Mathematische Funktionen/Klammerregeln . . . . .	86
2.2.1.6	Hexadezimal-(Sedezimal-)Zahlen . . . . .	87
2.2.1.7	Textausdrücke . . . . .	88
2.2.1.8	Logische Vergleichsausdrücke . . . . .	89
2.2.2	Sprachelemente . . . . .	91
2.2.2.1	Numerische Konstante . . . . .	91
2.2.2.2	Textkonstante . . . . .	91
2.2.2.3	Nummerische Variable . . . . .	91
2.2.2.4	Textvariable . . . . .	92
2.2.2.5	Numerische Funktionen, Textfunktionen . . . . .	92
2.2.2.6	Numerischer Ausdrücke . . . . .	93
2.2.2.7	Textausdrücke . . . . .	93
2.2.2.8	Logische Vergleichsausdrücke . . . . .	94
2.2.3	Variablen . . . . .	96
2.2.3.1	Standardvariable . . . . .	97
2.2.3.2	Einfache Variable . . . . .	98
2.2.3.3	Indizierte Variable . . . . .	98
2.2.3.4	Besonderheiten der Variablen A . . . . .	99
2.2.3.5	Feldvariablen . . . . .	101
2.2.3.6	Numerische Feldvariablen . . . . .	101
2.2.3.7	Textfeldvariablen . . . . .	102
2.2.4	Arithmetische Funktionen . . . . .	104
2.2.5	Textfunktionen . . . . .	105
2.2.6	Programmieren in BASIC . . . . .	106
2.2.6.1	BASIC-Übersicht . . . . .	106
2.2.6.2	Programmerstellung . . . . .	107
2.2.6.3	Programmaufbau . . . . .	108
2.2.6.4	Eingabe eines Programms . . . . .	108
2.2.6.5	Korrektur einer Zeile . . . . .	110
2.2.6.6	Löschen einer Zeile . . . . .	111
2.2.6.7	Programmausführung . . . . .	111
2.2.6.8	Fehlermeldungen/Fehlersuche . . . . .	112
2.3	BASIC REFERENZTEIL . . . . .	113
2.3.1	Kommandos . . . . .	116
2.3.2	Befehle . . . . .	130
2.3.3	Funktionen . . . . .	178

2.3.3.1	Pseudovariable . . . . .	178
2.3.3.2	Numerische Funktionen . . . . .	181
2.3.3.3	String-Funktionen . . . . .	188
2.3.4	Ein-/Ausgabe-Kommandos . . . . .	191
2.3.5	Text-Funktionskommandos . . . . .	208
2.4	FEHLERSUCHE . . . . .	211

**ANHANG**

A.	Fehlermeldungen . . . . .	216
B.	ASCII-Code Tabelle . . . . .	219
C.	Formatieren der Datenausgabe . . . . .	221
D.	Bewertung von Ausdrücken und Operatoren-Vorrang . . . . .	226
E.	Tastenfunktionen . . . . .	228
F.	Signale des E/A-Anschlusses . . . . .	233
G.	Technische Daten . . . . .	234
H.	Die Benutzung von Programmen, die für andere PC-Modelle entwickelt wurden . . . . .	236
	PROGRAMMBEISPIELE . . . . .	243
	INDEX . . . . .	287



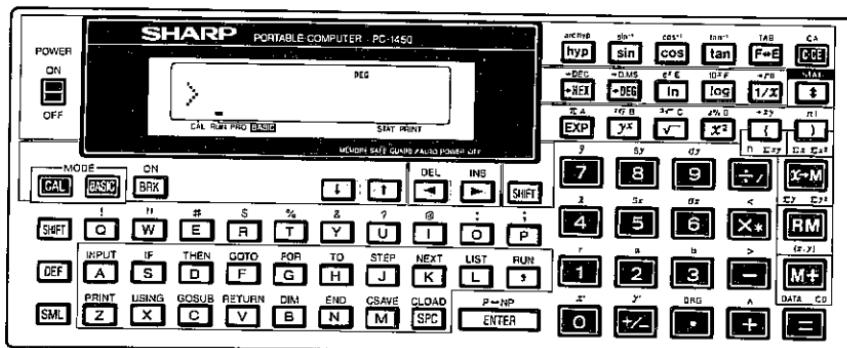
## **EINLEITUNG**

## EINLEITUNG

## I ALLGEMEINES

Mit dem **PC-1450** stellt SHARP einen preisgünstigen und dennoch außerordentlich leistungsstarken **BASIC-Taschencomputer** mit integriertem wissenschaftlichem Rechner vor.

Bitte beachten Sie, daß zum Arbeiten mit dem Computer eine RAM-Karte erforderlich ist. Wenn der Computer eingeschaltet wird, ohne daß eine RAM-Karte eingesetzt ist, funktioniert keine der Tasten.



## Frontansicht des PC-1450

Zusammen mit dem Thermodrucker mit integriertem Kassetten-Interface **CE-126P** oder dem Drucker **CE-140P** wird der **PC-1450** zu einer kleinen, leistungsfähigen Datenverarbeitungsanlage.

Bei der Gestaltung dieser Bedienungsanleitung wurde versucht, sowohl den Ansprüchen eines Anfängers als auch denen eines geübten Programmierers gerecht zu werden. In manchen Fällen war eine Kompromißlösung aber nicht zu umgehen, und wir verweisen daher bereits an dieser Stelle auf die umfangreiche BASIC-Literatur, die im Fachhandel erhältlich ist.

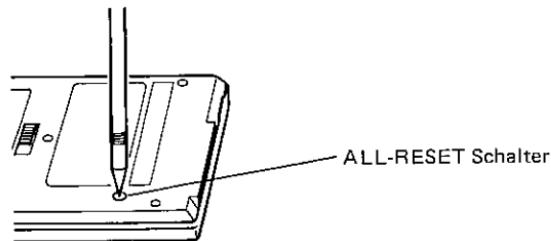
Wir hoffen, Ihnen mit diesem Handbuch alle möglichen Hilfsmittel in die Hand zu geben, so daß Sie Ihren PC-1450 für Ihre Belange optimal einsetzen können.

## II BETRIEBSHINWEISE

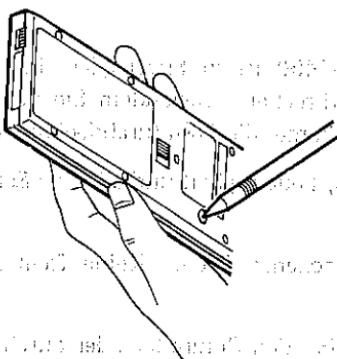
1. Die Flüssigkeitskristallanzeige des PC-1450 ist in Spezialglas eingebettet und kann daher bei Gewalteinwirkung zerbrechen. Behandeln Sie den Computer deshalb mit Sorgfalt. Beim Transport immer die Tastaturabdeckung verwenden.
2. Schützen Sie den Computer vor Staub, Feuchtigkeit und allzu großen Temperaturschwankungen.
3. Zur Reinigung dient ein weiches, trockenes Tuch. Keine Reinigungs- oder Lösungsmittel verwenden.
4. Durch elektrostatische Entladungen über den Computer oder durch Fehlbedienung (der Computer blieb beim Batteriewechsel oder Anschluß der Option CE-126P eingeschaltet) kann der Computer "abstürzen". Dadurch werden alle Tastenfunktionen einschließlich der [CE]-Taste blockiert.

Sollte Ihnen dies passieren, müssen Sie den Computer PC-1450 initialisieren. Dazu stehen zwei Möglichkeiten zur Auswahl:

**ALL RESET:** Reset-Schalter. Er wird gebraucht, um den Computer zu initialisieren, wenn CLEAR ( [CLS] ) oder [CA] nicht ausreichen, um das Problem zu lösen.



**ACHTUNG:** Um den Computer zu initialisieren, drücken Sie irgendeine Taste, halten diese fest und drücken gleichzeitig den ALL-RESET-Schalter auf der Rückseite des Gerätes. Die Programme und Variablen werden dabei nicht gelöscht.



Um die Programme und Daten auf der RAM-Karte zu löschen, ist es erforderlich, den ALL-RESET-Schalter zu betätigen. Dieser Schalter befindet sich unterhalb des Rechner-gehäuses, an der unteren rechten Seite. Um diesen Schalter zu betätigen, muß man mit einem spitzen Gegenstand, z.B. einem Kugelschreiber, auf die entsprechende Stelle drücken. Es kann vorkommen, daß der Schalter nicht sofort reagiert. In diesem Fall muß man ihn länger drücken. Wenn der Schalter funktioniert, wird er leicht eingesunken. Dies ist ein normaler Zustand.

**Zum Drücken des ALL-RESET-Schalters benutzen Sie bitte einen spitzen Gegenstand, z.B. einen Kugelschreiber. Vermeiden Sie Gegenstände, deren Spitzen abbrechen könnten (z.B. Bleistifte!).**

Sollte Ihr Computer nach dieser Operation immer noch nicht reagieren, betätigen Sie den ALL-RESET-Schalter, ohne dabei eine andere Taste zu drücken. Dabei werden allerdings Daten und Programme gelöscht – man sollte also den ALL-RESET-Schalter allein nur betätigen, wenn es unumgänglich ist.

Wenn das Gerät anschließend noch nicht normal arbeitet, entfernen Sie bitte die Batterien. Nach 10 Sekunden setzen Sie die Batterien wieder ein und betätigen nochmals den ALL-RESET-Schalter.

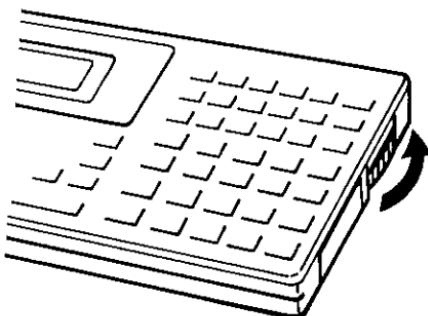
## FUNKTIONEN DES ALL-RESET-SCHALTERS

Wird nur der ALL-RESET-Schalter betätigt, werden das Programm und alle Daten gelöscht.

Ist zu diesem Zeitpunkt eine RAM-Karte in der Benutzung, werden ebenso Programme und Daten auf der RAM-Karte gelöscht.

Wird bei Betätigung des ALL-RESET-Schalters gleichzeitig eine Taste gedrückt, so werden die Programme und Daten beibehalten. Seien Sie vorsichtig beim Drücken des ALL-RESET-Schalters.

## KONTRASTKNOPF



**Kontrastknopf:** Durch Drehen des Kontrastknopfes in Pfeilrichtung wird der Kontrast der Anzeige verstrkrt, in entgegengesetzter Richtung verringert.

## III. STROMVERSORGUNG

### BATTERIEWECHSEL

Der PC-1450 arbeitet ausschlielich mit Lithium-Batterien. Wenn er an die Option CE-126P angeschlossen ist, kann der Computer auch uber diese versorgt werden, sofern deren momentane Betriebsspannung hoher ist. Dies verringert den Stromverbrauch der Lithium-Zellen.

Beim Wechseln der Batterien beachten Sie bitte unbedingt die folgenden Hinweise:

- Wechseln Sie grundsatzlich beide Batterien gleichzeitig.
- Verwenden Sie niemals eine neue gemeinsam mit einer gebrauchten Batterie.
- Benutzen Sie nur Lithium-Zellen (Typ CR-2032).

### EINSETZEN DER BATTERIEN

Wenn die Anzeige bei maximal eingestelltem Kontrast schwach und schlecht zu erkennen ist, zeigt dies an, da die Batterien verbraucht sind. In diesem Fall mussen Sie die Batterien umgehend erneuern.

- (1) Schalten Sie den Computer ab, indem Sie den Schiebschalter in die OFF-Position bringen.
- (2) Entfernen Sie den ruckwartigen Gehusedeckel, indem Sie den Verriegelungsknopf (siehe Abbildung 1) in Pfeilrichtung schieben.

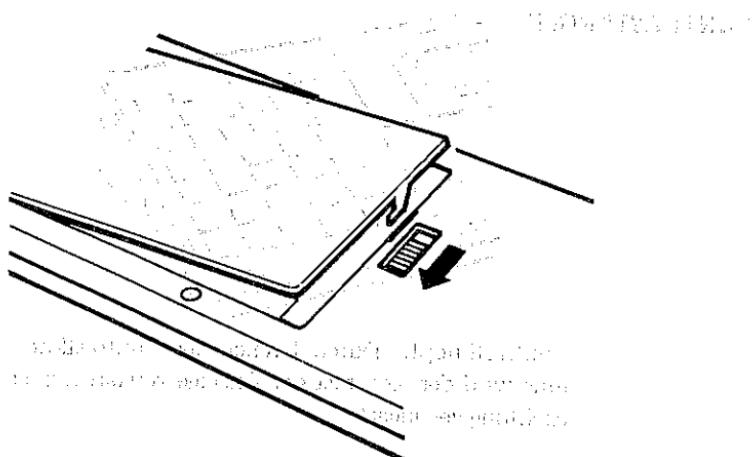


Abb. 1

- (3) Wenn Sie eine RAM-Karte benutzen, entfernen Sie diesen wie in Abschnitt 1.2 beschrieben.
- (4) Schieben Sie die Batterie-Abdeckung in Pfeilrichtung, sie lässt sich dann abheben.

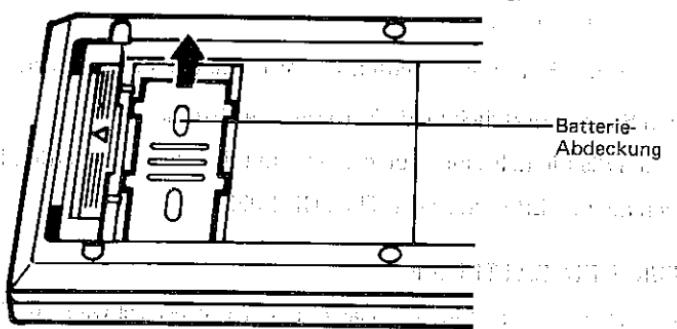


Abb. 2

Die RAM-Karte kann leicht aus dem Steckplatz entnommen werden. Dazu ist es erforderlich, die entsprechende Schraube zu lösen und die Karte nach unten zu drücken. Die Karte kann dann leicht aus dem Steckplatz entnommen werden.

- (5) Erneuern Sie beide Batterien.

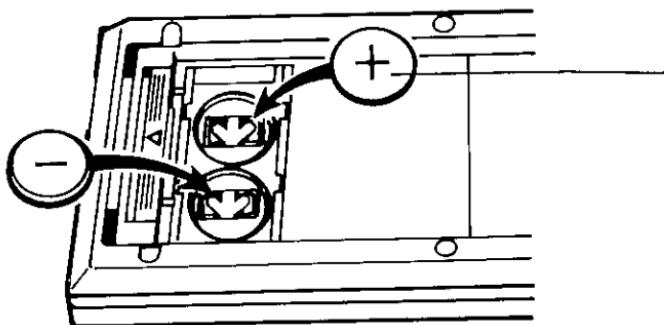
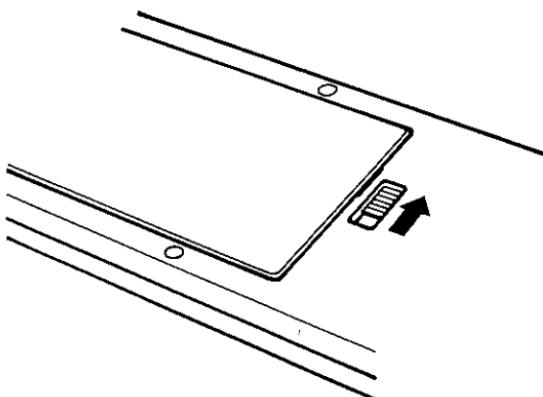


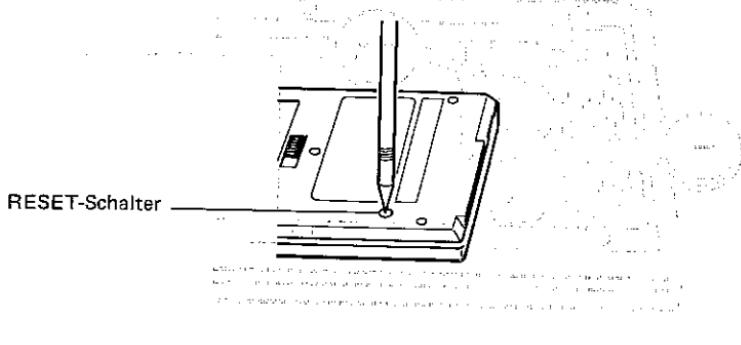
Abb. 3

- (6) Setzen Sie die Batterie-Abdeckung durch Schieben gegen die Pfeilrichtung wieder ein.  
(7) Setzen Sie den rückwärtigen Gehäusedeckel wieder auf und schieben Sie den Verriegelungsknopf wieder in seine ursprüngliche Position.

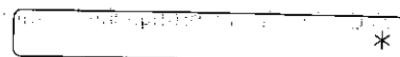


Beachten Sie: Vergewissern Sie sich, daß der Riegel in der LOCK-Position ist:  
Der Computer arbeitet sonst nicht. Wenn Sie den Computer einschalten, ohne daß der Riegel in der LOCK-Position ist, schieben Sie den Riegel in die LOCK-Position, schalten den Computer wieder aus und anschließend wieder ein.

- (8) Bringen Sie den Schiebeschalter in die Stellung "ON" und betätigen Sie den RESET-Schalter, um den Computer zu initialisieren.



- (9) Das Display muß wie folgt aussehen:



Sollte keine Anzeige auf dem Display erscheinen oder sollten andere Symbole als "\*" angezeigt werden, entnehmen Sie die Batterien und setzen sie wieder ein und überprüfen danach die Anzeige erneut.

- (10) Den Computer ausschalten und die RAM-Karte einsetzen. (Siehe Abschnitt 1.2.)

#### Hinweis:

Nach Austauschen der Batterien müssen vor Einsetzen der RAM-Karte unbedingt die Schritte (8) und (9) ausgeführt werden. Wird die RAM-Karte eingesetzt, während in Schritt (8) der Computer eingeschaltet ist, wird der Inhalt der RAM-Karte gelöscht.

#### Referenz:

Die Hauptbatterie kann ausgetauscht werden, während die RAM-Karte CE-211M im Computer eingesetzt ist. Dafür den Rückstellknopf an der Rückseite in Schritt (8) drücken, bevor der Computer eingeschaltet wird. BEIM DRÜCKEN DES RESET-SCHALTERS DARF DER COMPUTER AUF KEINEN FALL EINGESCHALTET SEIN. Wenn der Computer eingeschaltet ist, wird der Inhalt der RAM-Karte gelöscht.

#### • Austauschen der Batterie für RAM-Karte CE-211M

Nach Einsetzen einer Batterie in die RAM-Karte bleibt der Inhalt der RAM-Karte für die folgenden Zeitspannen erhalten. Die angegebenen Zeiten gelten

für eine Lagertemperatur von 25°C.

- Ca. 5 Jahre, wenn die RAM-Karte im Computer eingesetzt ist.
- Ca. 24 Monate, wenn die RAM-Karte aus dem Computer entnommen ist.

Bitte denken Sie daran, vor Ablauf der oben genannten Zeitdauern eine neue Batterie einzusetzen. Schreiben Sie das Datum, an dem Sie die neue Batterie einsetzen, in die Zeile neben dem Batteriefach, so daß Sie wissen, wann Sie die Batterie austauschen müssen.

Sehr hohe oder niedrige Temperaturen können die Lebensdauer der Batterie verkürzen, so daß der Inhalt der RAM-Karte schon vor Ablauf der oben angegebenen Zeitdauern gelöscht wird. In den folgenden Fällen ist das Austauschen der Batterien erforderlich.

1. Programme im Speicher können nicht ausgeführt werden.
2. Auf dem Display erscheinen bedeutungslose Daten und Anzeigen.
3. Es treten häufig Fehler ohne erkennbare Ursache auf.

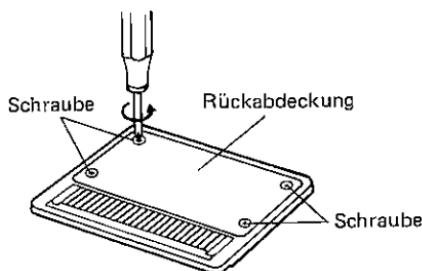
### Auswechseln der Batterie

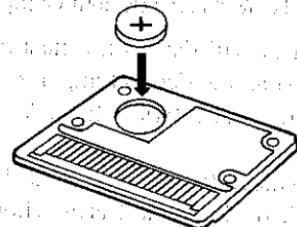
#### Hinweis:

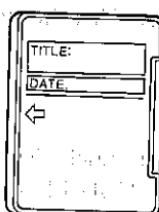
Vor Auswechseln der Batterie sollte der Inhalt der RAM-Karte auf Cassette aufgenommen werden. ALLE DATEN UND PROGRAMME WERDEN GE-LÖSCHT, WENN DIE BATTERIE AUS DER RAM-KARTE ENTNOMMEN WIRD.

- Die Batterie der RAM-Karte CE-201M/CE-202M (Option) kann ausgetauscht werden, während die RAM-Karte im Computer eingesetzt ist. Dann wird der Inhalt der RAM-Karte durch die Spannungsversorgung vom Computer erhalten.

- (1) Die vier Schrauben mit einem Kreuzschlitzschraubenzieher entfernen und die Rückabdeckung entfernen.



- (2) Die alte Batterie entnehmen. Die neue Batterie mit einem weichen, trockenen Tuch abwischen und einsetzen.
- 
- (3) Die Rückabdeckung wieder anbringen und die Schrauben anziehen.
- (4) Das Datum mit unlösbarer Tinte in die Zeile DATE schreiben.



- Die Namen der in der RAM-Karte gespeicherten Programme in die Zeile TITLE schreiben.
- Die RAM-Karte wieder in den Computer einsetzen und die zwischenzeitlich auf Cassette gespeicherten Programme einlesen.

**Achtung:** Läßt man eine leere Batterie im Computer, kann er durch Säureausritt aus der Batterie geschädigt werden. Leere Batterien umgehend erneuern!

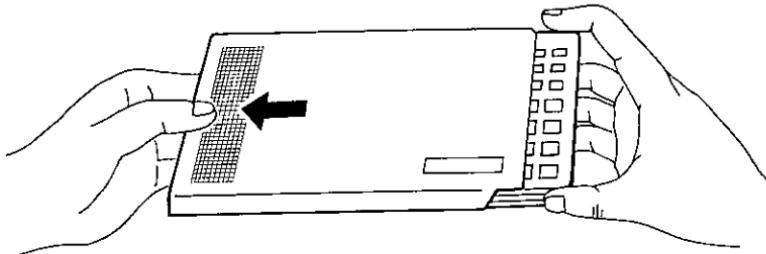
**Vorsicht:** Bewahren Sie die Batterien außerhalb der Reichweite von Kindern auf!

#### IV. TASTATURABDECKUNG

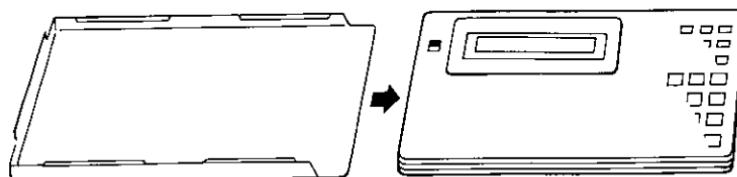
Zum Schutz der Tastatur und der Anzeige ist der PC-1450 mit einer stabilen, doppelseitig aufschiebbaren Tastaturabdeckung ausgestattet.

Die Tastaturabdeckung des Computers wie dargestellt verwenden.

Bei Gebrauch:

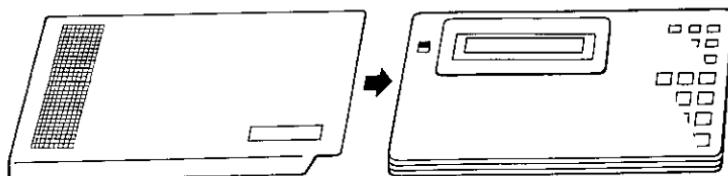


Die Tastaturabdeckung ist so gestaltet, daß sie während der Verwendung des Rechners auf dessen Unterseite geschoben werden kann.



Tastaturabdeckung bei Gebrauch des Rechners

Nach Gebrauch:



Tastaturabdeckung nach Gebrauch des Rechners

## V. RICHTIGE BEHANDLUNG DES COMPUTERS

Um reibungsloses Funktionieren Ihres Computers zu gewährleisten, empfehlen wir die Beachtung der folgenden Punkte:

- Gehen Sie immer vorsichtig mit Ihrem Taschencomputer um, da die Flüssigkristallanzeige aus Glas gefertigt ist.
- Halten Sie den Computer von starken Temperaturschwankungen fern, ebenso von Feuchtigkeit oder Staub. Wenn Sie bei warmen Wetter Ihren PKW längere Zeit der direkten Sonne aussetzen und sich hohe Temperaturen aufbauen, kann das Ihren Computer beschädigen!
- Benutzen Sie zur Reinigung Ihres Computers ausschließlich ein trockenes, weiches Tuch. Verwenden Sie niemals Lösungsmittel, Wasser oder ein feuchtes Tuch!
- Um das Auslaufen der Batterien zu vermeiden, entfernen Sie diese, wenn Sie den Computer für längere Zeit nicht benutzen wollen.
- Wenn Sie die Hilfe einer Werkstatt benötigen, geben Sie Ihren Computer nur in ein autorisiertes SHARP Service Center oder wenden Sie sich an einen SHARP Fachhändler.
- Wenn der Computer starker statischer Aufladung oder auch starken Störungen ausgesetzt ist, kann er sich unter Umständen "aufhängen" (d. h. auf Tastendruck nicht mehr reagieren). Falls dies auftritt, drücken Sie den ALL-RESET-Schalter und halten dabei eine Taste fest (siehe Pannenhilfe).
- Bewahren Sie dieses Handbuch gut auf, falls Sie später einmal etwas nachschlagen wollen.

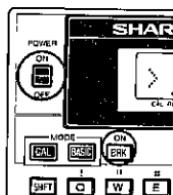
**TEIL I**  
**THEORIE**

## TEIL I THEORIE

### 1.1 GRUNDLAGEN SHARP PC-1450

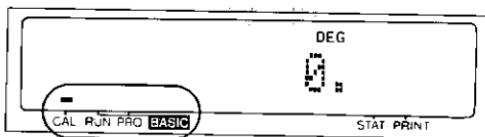
#### 1.1.1 EINSCHALTEN DES COMPUTERS

Das Einschalten des Computers erfolgt zum einen über den Schiebeschalter links neben dem Display, zum anderen über die Taste **ON BRK**.



Nach dem Einschalten erscheinen auf der Anzeige folgende Symbole:

Zeigt die Winkeleinheit (DEG/GRAD/RAD) an.



Zeigt Betriebsart (CAL/RUN/PRO)

#### 1.1.2 AUSSCHALTEN DES COMPUTERS

##### 1.1.2.1 Automatisch

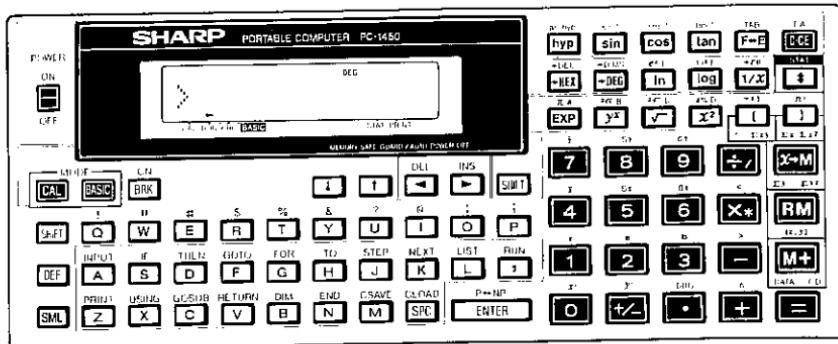
Der Computer schaltet sich nach ca. 10 Minuten nach der letzten Tastenbedienung automatisch ab. Dabei bleiben alle Programm- und Dateninformationen gespeichert.

Hat sich der Rechner automatisch ausgeschaltet, muß er über die Taste **ON BRK** wieder aktiviert werden. Auf der Anzeige erscheint die zuletzt bearbeitete oder angezeigte Zeile.

##### 1.1.2.2 Manuell

Soll der Computer sofort nach Gebrauch ausgeschaltet werden, so muß der Schiebeschalter links neben dem Display in Stellung OFF gebracht werden. Die Programm- und Dateninformationen bleiben dabei gespeichert. Der Computer läßt sich auch während einer Programmausführung abschalten.

### 1.1.3 BEDIENELEMENTE



Das Bedienfeld des PC-1450 besteht aus 78 Tasten, die zu 4 Blöcken zusammengefaßt sind:

- die Schreibmaschinen-Alphatastatur mit **ENTER**-Taste (2 Ebenen)
- die Tasten für die Sonderfunktionen
- die numerische Tastatur mit den Grundrechnungsarten (2 Ebenen)
- die Tasten für die wissenschaftlichen Funktionen (2 bzw. 3 Ebenen)

Die Schreibmaschinen-Alphatastatur ist mit zwei Ebenen belegt. Bei einfachem Tastendruck gelten die auf den Tasten stehenden Bezeichnungen. Will man die über den Tasten stehenden Sonderzeichen (oberste Reihe der Tastatur) oder die vorprogrammierten BASIC-Anweisungen abrufen, muß vorher die Doppelfunktions-taste **SHIFT** gedrückt werden.

Mit der Taste **DEF** kann den unteren beiden Reihen noch eine dritte Ebene zugewiesen werden. Über die sogenannten 'Definable Keys' können verschiedene Programme sofort gestartet werden; man drückt dazu die Taste **DEF** und eine dem Programm zugewiesene Taste.

In der Reihe über der Alpha-Tastatur befinden sich die Sonderfunktionstasten zur Wahl der Betriebsart, zum Unterbrechen oder zum Auflisten sowie zum Editieren eines Programms.

Rechts neben der Alpha-Tastatur befindet sich die numerische Tastatur. Die zweite Ebene der numerischen Tastatur, die hauptsächlich für statistische Berechnungen dient, ist ebenfalls mit der **SHIFT**-Taste erreichbar.

Mit den Tasten oberhalb der numerischen Tastatur lassen sich die meisten wissenschaftlichen Berechnungen, wie trigonometrische Berechnungen, Hyperbelberechnungen, Polarkoordinatenberechnungen usw. durchführen. Dieser Teil der Tastatur ist mit bis zu drei Ebenen belegt.

Die **SHIFT**-, **BASIC**- und **hyp**-Tasten sind Wechselschalter, d.h. durch einmaliges Drücken wird in den jeweiligen Mode umgeschaltet, durch nochmaliges Drücken der Urzustand wieder hergestellt.

### 1.1.4 TASTENFUNKTIONEN

- [DEF]** In Verbindung mit den Tasten A, S, D, F, G, H, J, K, L, ;, Z, X, C, V, B, N, M, SPC ermöglicht diese Taste das Starten des Programms über einen Markennamen.
- [SHIFT]** Doppelfunktionstaste
- [↑] [↓]** Editertasten
- [←]** Cursor nach links; Editieraufzug
- [→]** Cursor nach rechts; Editieraufzug
- [ON BRK]** Unterbrechen des Programmablaufs. Am Display erscheint die Meldung **BREAK IN XXX** (XXX = Zeilennummer). Unterbricht auch Drucker- und Cassettenbetrieb.
- C-C.E.** Löscht die Anzeige und die Fehlermeldungen.
- [SHIFT] [INS]** Einfügen einzelner Zeichen (Platzhalter  wird gesetzt)
- [SHIFT] [DEL]** Löschen einzelner Zeichen
- [ON BRK]** Einschalten des Rechners nach automatischer Abschaltung.
- [CA] [C-C.E.]** Löschen der Anzeige und
- Löschen des WAIT-Intervalls
  - Löschen des USING-Formats
  - Aufheben des TRACE-Betriebs
  - Löschen der Fehlermeldungen
- [A] ~ [Z]** Alphastatur A–Z; Variablennamen
- [SPC]** Leerzeichen
- [ENTER]** Beschließt die Eingabe einer Programmzeile. Beim Rechnen ohne Programmunterstützung im RUN-Modus ersetzt die Taste das „=“-Zeichen.
- Programmstart in Verbindung mit RUN oder GOTO**
- [~] [S]** Sonderzeichen
- [SHIFT] [A] ~** Aufruf der wichtigsten BASIC-Anweisungen
- [SHIFT] [SPC]**

<b>SHIFT</b>	<b>P+NP</b>	Umschaltung für Druck/Nichtdruck beim Rechnen ohne Programmunterstützung im RUN/PRO-Modus
<b>.</b>		Dezimalpunkt
<b>0</b>	$\sim$	<b>9</b> numerische Zeichen
<b>+</b>	$\sim$	<b><math>\frac{1}{x}</math></b> Grundrechnungsarten
<b>hyp</b>		Umschaltung Hyperbelfunktionen
<b>SHIFT</b>	<b>archyp</b>	Umschaltung Umkehrfunktionen der Hyperbelfunktionen
<b>sin</b>		Trigonometrische Funktionen
<b>SHIFT</b>	<b>sin<sup>-1</sup></b>	Umkehrfunktionen der trigonometrischen Funktionen
<b>F<sub>fE</sub></b>		Umschaltung Fließkomma/Festkomma
<b>SHIFT</b>	<b>TAB</b>	Festlegung der Nachkommastellen
<b>HEX</b>		Umrechnung Dezimalsystem $\leftarrow$ Hexadezimalsystem
<b>SHIFT</b>	<b>DEC</b>	Umrechnung Hexadezimalsystem $\leftarrow$ Dezimalsystem
<b>D-DEG</b>		Umrechnung Dezimalsystem $\leftarrow$ Sexagesimalsystem
<b>SHIFT</b>	<b>D-MIN</b>	Umrechnung Sexagesimalsystem $\leftarrow$ Dezimalsystem
<b>ln</b>		Natürlicher Logarithmus
<b>SHIFT</b>	<b>e<sup>x</sup></b>	Exponentialfunktion $e^x$
<b>log</b>		Zehnerlogarithmus
<b>SHIFT</b>	<b>10<sup>x</sup></b>	Exponentialfunktion $10^x$
<b>1/x</b>		Reziprokerwert
<b>SHIFT</b>	<b>xy</b>	Umwandlung Polarkoordinaten in rechtwinklige Koordinaten
<b>SHIFT</b>	<b>rθ</b>	Umwandlung rechtwinklige Koordinaten in Polarkoordinaten
<b>EXP</b>		Exponentielldarstellung
<b>SHIFT</b>	<b>π</b>	Konstante Pi
<b>y<sup>x</sup></b>		Potenzierfunktion
<b>SHIFT</b>	<b>x<sup>y</sup></b>	x-te Wurzel
<b><math>\sqrt{x}</math></b>		Quadratwurzelfunktion
<b>SHIFT</b>	<b>3<math>\sqrt{x}</math></b>	Dritte Wurzel
<b><math>x^2</math></b>		Quadratzahlfunktion

<b>[SHIFT]</b>	<b>[n!]</b>	Fakultät
<b>[{ }]</b>		Klammern
<b>X=M</b>	<b>~ M+</b>	Speicherrechnung
<b>[SHIFT]</b>	<b>DRG</b>	Umschaltung der Winkeleinheit
<b>[SHIFT]</b>	<b>&gt;</b>	Logische Vergleichsoperatoren
<b>[SHIFT]</b>	<b>&lt;</b>	
<b>[SHIFT]</b>	<b>STAT</b>	Statistikbetriebsart

### 1.1.5 DIE ANZEIGE

Der **PC-1450** verfügt über eine 16-stellige alphanumerische Flüssigkristallanzeige. Die einzelnen Zeichen werden in einer 5 x 7-Punktmatrix dargestellt.

Das Eingaberegister faßt 80 Zeichen. Alle Zeichen, die über die 16 möglichen darstellbaren Zeichen der Anzeige hinausgehen, müssen mit der Cursor-Tasten (**►** und **◄**) 'gescrolled' werden.

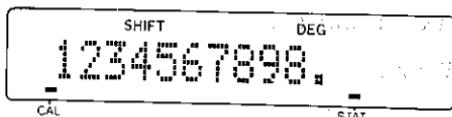
Der Cursor zeigt die Stelle an, an der die nächste Eingabe erfolgen kann. Er steht normalerweise auf der ersten freien Stelle und wird durch einen kurzen waagrechten Strich (**\_**) dargestellt.

Eine Ausnahme besteht, wenn auf der Anzeige links das Bereitschaftssymbol (**>**) angezeigt wird. Durch das nächste einzugebende Zeichen wird dieses überschrieben. Erst dann sieht man den Cursor auf der ersten freien Stelle nach dem zuletzt eingegebenen Zeichen.

Stellt man den Cursor mit Hilfe der **►**- oder der **◄**-Taste über ein bereits eingegebenes Zeichen, so erkennt man die Position am Blinken des Cursors an dieser Stelle in der vollen Punktmatrix.

Anhaltender Druck auf die Cursor-Tasten läßt den Cursor schnell an die gewünschte Stelle der Zeile bringen. Ein kurzer Druck verschiebt die Anzeige nur um ein Zeichen.

Versucht man mehr als 80 Zeichen einzulesen, werden diese vom Rechner nicht mehr angenommen. Jede zusätzliche Eingabe überschreibt das zuletzt eingegebene Zeichen. Der Cursor verändert hier sein Aussehen; er blinkt auf dem letzten Zeichen in der vollen Punktmatrix.

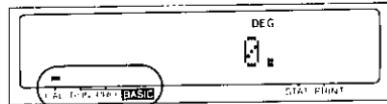


### 1.1.6 WAHL DER BETRIEBSART

Beim PC-1450 werden drei Betriebsarten unterschieden:

1. CAL-Mode: Der Computer wird als wissenschaftlicher Rechner verwendet.
2. RUN-Mode: Gespeicherte Programme können gestartet werden. Zusätzlich kann der Computer im RUN-Mode auch als gewöhnlicher Taschenrechner ohne Programmunterstützung verwendet werden.
3. PRO-Mode: BASIC-Programme können eingegeben, gelistet und verändert werden.

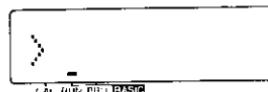
Die Wahl der Betriebsart erfolgt mit den grünen Tasten **CAL** und **BASIC**. Die gewählte Betriebsart wird am Display durch einen kleinen waagrechten Strich (-) angezeigt.



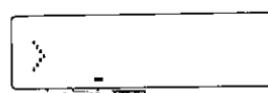
Wird der Computer eingeschaltet, so ist automatisch die Betriebsart CAL (culation) gewählt. Im Display wird das wie dargestellt angezeigt:



Wenn Sie nun die grüne Taste **BASIC** drücken, so wird die Betriebsart in den RUN-Mode umgeschaltet.



Durch nochmaliges Drücken der Taste **BASIC** wird in den PRO-Mode umgeschaltet.



Die **BASIC**-Taste ist ein Wechselschalter, d.h., durch einmaliges Drücken wird die Betriebsart umgeschaltet, durch nochmaliges Drücken der Urzustand wieder hergestellt.

Durch Drücken der **CAL**-Taste kann der Rechner wieder in den CAL-Mode umgeschaltet werden.

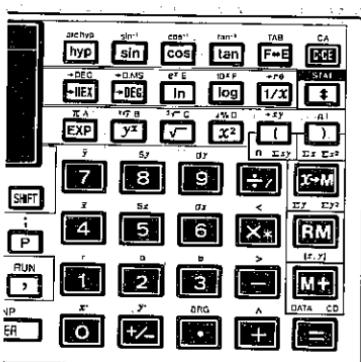
### 1.1.6.1 CAL-Mode

Schalten Sie Ihren Rechner durch Drücken der **CAL**-Taste in den CAL-Mode.

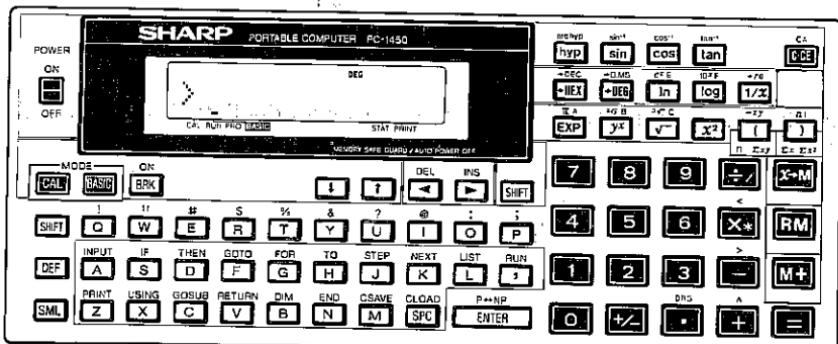
In diesem Mode können Sie mit den rechts abgebildeten Tasten den PC-1450 als wissenschaftlichen Rechner verwenden.

**Beispiel:**

- |                   |       |
|-------------------|-------|
| <b>C-CE</b>       | → 0.  |
| <b>1</b> <b>2</b> | → 12. |
| <b>+</b>          | → 12. |
| <b>3</b>          | → 3.  |
| <b>=</b>          | → 15. |



### 1.1.6.2 RUN- und PRO-Mode



Schalten Sie Ihren Rechner durch Drücken der Taste **BASIC** in den RUN- bzw. in den PRO-Mode. In dieser Betriebsart können die oben abgebildeten Tasten zur Eingabe verwendet werden.

**Beispiele:**

<b>PRINT</b>	<b>USING</b>	<b>GOSUB</b>	<b>C</b>	→ ZXC _
<b>1</b>	<b>2</b>	<b>DRG</b>	<b>b</b>	→ ZXC12.3 _
				→ >
<b>INPUT</b>	<b>A</b>	<b>=</b>	<b>4</b>	→ A = 4 + 5 _
			<b>+</b>	↑ Cursor
<b>CA</b>	<b>C-CE</b>	<b>PRINT</b>	<b>Z</b>	→ PRINT _
<b>SHIFT</b>	<b>W</b>	<b>(</b>	<b>)</b>	→ PRINT "(√ _

**1.2 DIE RAM-KARTE**

Der Computer speichert alle Programme und Daten (feste Variable, einfache Variable, Feldvariable usw.) in der RAM-Karte (RAM = Random Access Memory = Direktzugriffsspeicher). Beim Kauf des Computers ist die RAM-Karte CE-211M bereits installiert. Die RAM-Karten CE-201M und CE-202M sind als Sonderzubehör erhältlich und können einfach in den Computer eingesetzt werden. Die RAM-Karten CE-211M, CE-201M und CE-202M verfügen über eine Speicherkapazität von 4 KBytes, 8 KBytes bzw. 16 KBytes.

Durch die Verwendung einer RAM-Karte größerer Kapazität kann der Speicherbereich für Programme und Daten einfach und schnell erweitert werden. Außerdem können auf RAM-Karten Programme gespeichert werden, die gegenwärtig nicht benutzt werden.

Das Programm in der RAM-Karte bleibt auch dann erhalten, wenn die Karte aus dem Computer entnommen wird, da die Karte mit einer eigenen Batterie ausgerüstet ist. Deshalb kann das Programm zusammen mit der Karte erneut in den Computer "eingeladen" werden, indem Sie einfach die Karte wieder in den Rechner einstacken.

**Kapazitäten des Programm-/Datenspeichers**

CE-211M	CE-201M	CE-202M
3070	7166	15358

\* Die Entnahme der Batterie aus der RAM-Karte löscht die in der Karte enthaltenenen Daten und Programme. Deshalb sollte ein Programm aus der RAM-Karte erst auf ein Band überspielt werden, bevor die Batterie gewechselt wird.

Wird die Batterie der RAM-Karten CE-201M und CE-202M gewechselt, während die Karte in den Computer eingesetzt ist, so bleiben alle Daten und Programme der Karte erhalten.

Die Daten und Programme werden von einer Batterieladung für etwa 24 Monate bei der CE-211M, 34 Monate bei der CE-201M und etwa 18 Monate bei der CE-202M gehalten. Die Zeiten, in denen die Karte im Computer eingesetzt ist, zählen dabei nicht mit.

Achtung: Drücken Sie nie den ALL RESET Schalter an der Rückseite des Computers, wenn eine Karte eingesetzt ist, weil dann alle Daten und Programme, die in der RAM-Karte gespeichert sind, gelöscht werden. Sollte es trotzdem nötig sein, den ALL RESET Schalter zu drücken, so lesen Sie bitte die entsprechenden Kapitel im Handbuch, wie die Daten der Karte erhalten bleiben.

### 1.2.1. Installation der RAM-Karte (CE-201M, CE-202M)

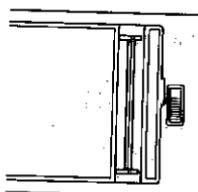
Wird die RAM-Karte das erste Mal eingesetzt, sollten Sie darauf achten, daß die Batterie eingesetzt ist.

#### (1) Schalten Sie Ihren Computer aus.

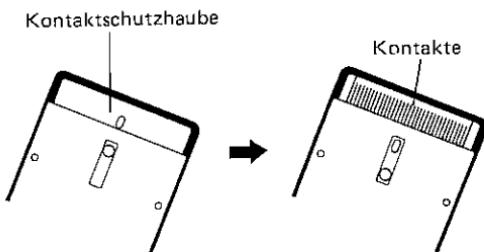
Der Computer muß immer abgeschaltet sein, wenn die RAM-Karte eingesetzt oder herausgenommen werden soll:

#### (2) Schieben Sie die Arretierung der Klappe auf der Rückseite des Rechners in Pfeilrichtung und entfernen Sie die Klappe.

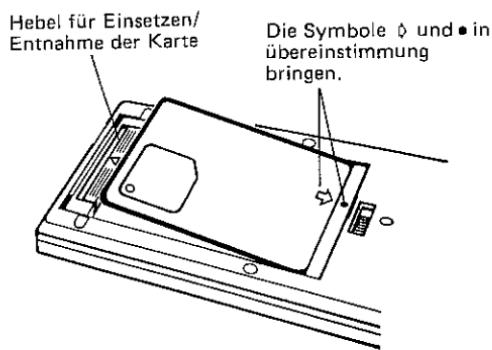
Achten Sie bitte darauf, keine Bauteile oder Kontakte im Inneren des Computers zu berühren.



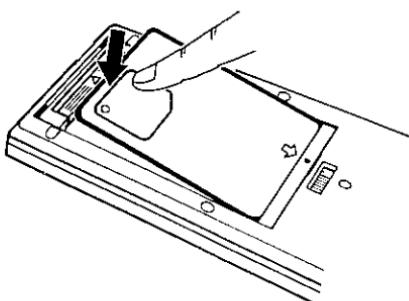
- (3) Öffnen Sie die Kontaktenschutzhäube der RAM-Karte. Achten Sie bitte darauf, daß die Haube ganz geöffnet ist und berühren Sie keinen der Kontakte.



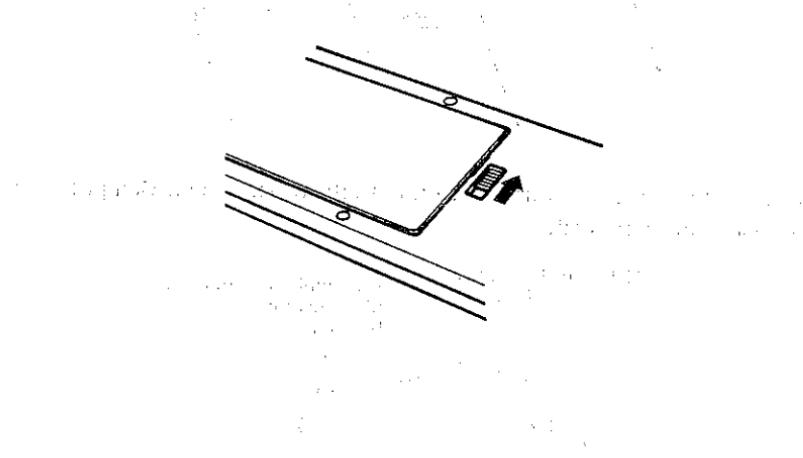
- (4) Schieben Sie die Karte mit der Kontaktseite voran in den Computer, wie in der Zeichnung dargestellt.



Schieben Sie die Karte nie anders herum in den Rechner und mit geschlossener oder nur teilweise geöffneter Schutzhäube. Sowohl die Karte als auch der Computer könnten zerstört werden. Drücken Sie dann auf das Gehäuse der Karte, bis sie einrastet.

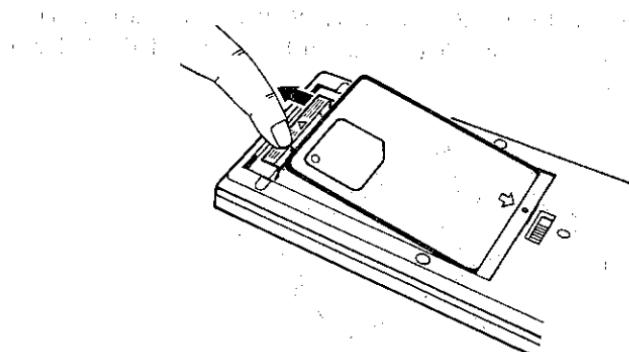


- (5) Setzen Sie die Klappe des Computers wieder ein und arretieren Sie sie, indem Sie den Hebel wieder in die Position "LOCK" schieben. Der Computer wird nicht arbeiten, wenn die Arretierung nicht ganz in die "LOCK"-Position geschoben wurde, selbst wenn alle Batterien korrekt eingesetzt sind. Sollte das Gerät nicht einwandfrei funktionieren, schalten Sie den Computer noch einmal ab, überprüfen die Arretierung der Klappe auf der Rückseite und schalten ihn dann wieder ein.



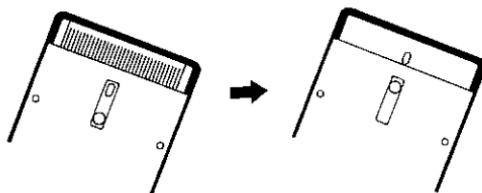
### 1.2.2. Entnahme der RAM-Karte (CE-201M, CE-202M)

- (1) Schalten Sie Ihren Computer aus.
- (2) Entfernen Sie die Klappe auf der Rückseite des Rechners wie oben beschrieben. Schieben Sie dann die Kartendarretierung in Pfeilrichtung, um die RAM-Karte zu lösen.

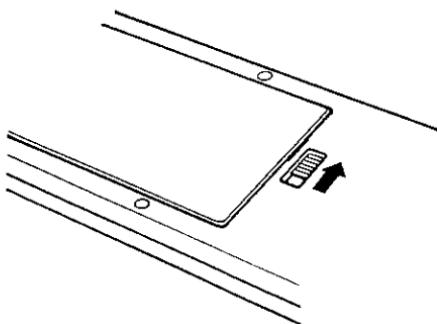


- (3) Schließen Sie bitte sofort die Kontaktschutzhäube der RAM-Karte, nachdem sie herausgenommen ist.

Achten Sie darauf, daß sie vollständig geschlossen ist und keine Kontakte mehr zu sehen sind.



- (4) Setzen Sie die Klappe des Computers wieder ein und schieben Sie die Arretierung wieder in die "LOCK"-Position.



### 1.2.3 Installation der RAM-Karte (CE-211M)

Wird die RAM-Karte das erste Mal eingesetzt, sollten Sie darauf achten, daß die Batterie eingesetzt ist.

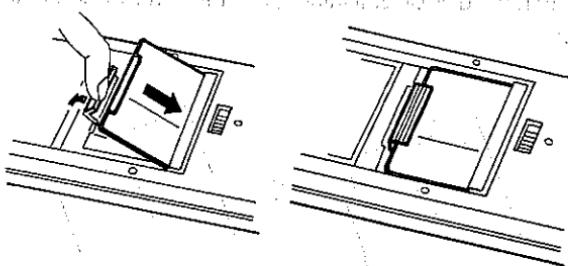
- (1) Schalten Sie den Computer aus.

Der Computer muß immer ausgeschaltet sein, wenn die RAM-Karte eingesetzt oder entnommen werden soll.

- (2) Schieben Sie die Arretierung der Klappe auf der Rückseite des Computers in Pfeilrichtung und entfernen Sie die Klappe.

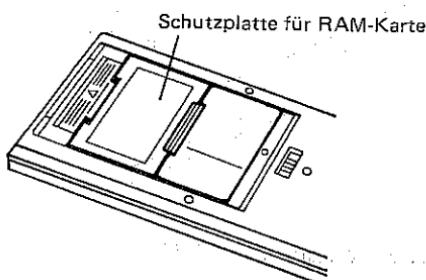
Achten Sie darauf, keine Bauteile oder Kontakte im Innern des Computers zu berühren.

- (3) Heben Sie die Feder an, die die RAM-Karte hält, und setzen die RAM-Karte aus dem Gehäuse. Setzen Sie dann die RAM-Karte mit den Kontakten nach unten in den Einschub ein.



Setzen Sie die Karte keinesfalls falsch herum ein, weil dadurch Kurzschlüsse entstehen und zu Beschädigungen von Computer und RAM-Karte führen könnten.

- (4) Setzen Sie die RAM-Karte-Schutzplatte ein, bringen die Rückabdeckung des Computers an und schieben die Arretierung auf die Position "LOCK". Die Arretierung muß unbedingt auf "LOCK" geschoben werden, weil sonst der Computer nicht arbeitet, auch wenn die Batterien richtig eingesetzt sind. Wird der Computer eingeschaltet, während die Arretierung nicht auf "LOCK" steht, schieben Sie die Arretierung auf "LOCK", schalten den Computer aus und dann wieder ein.



#### 1.2.4 Entnahme der RAM-Karte (CE-211M)

- (1) Schalten Sie den Computer aus.
- (2) Entfernen Sie die Rückabdeckung des Computers, entnehmen die Schutzplatte für die RAM-Karte und heben die Feder an, die die RAM-Karte hält. Entnehmen Sie dann die RAM-Karte. Bewahren Sie die RAM-Karte immer im Gehäuse der neu eingesetzten RAM-Karte auf.
- (3) Bringen Sie die Rückabdeckung wieder an und schieben die Arretierung auf "LOCK".

### 1.2.5 Verwendung der RAM-Karte mit anderen Modellen

Die RAM-Karten CE-211M, CE-201M und CE-202M, die mit diesem Computer verwendet werden können, können auch mit dem PC-1350 und/oder PC-2500 verwendet werden. Auch umgekehrt ist Austausch möglich.

#### (1) Programm-Austauschbarkeit

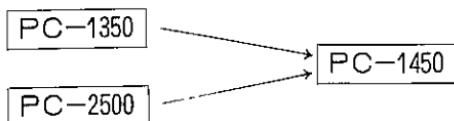


Nach Installation der RAM-Karte und Einschalten der Spannungsversorgung sollte der Schalter ALL RESET an der Rückseite des Computers gedrückt werden.

#### (2) Verwendung von Programmen

##### 1) Programmen, die auf anderen Modellen geschrieben wurden

Diese Programme können unverändert verwendet werden. Die folgenden Punkte müssen jedoch beachtet werden.



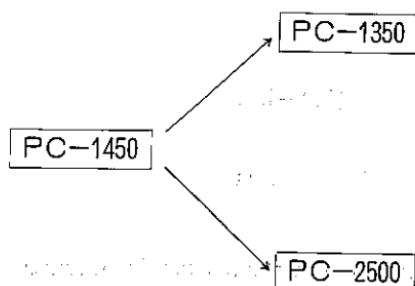
- Ein Veränderung des Programms ist erforderlich, wenn es Befehle enthält, die für diesen Computer nicht verfügbar sind.
- Vorher eingegebene Daten können nicht verwendet werden und werden gelöscht.
- Auf dem Display erscheinen Asterixe, siehe Abbildung unten, wenn das Programm zu groß für diesen Computer ist.



Wenn diese Asterixe erscheinen, schalten Sie den Computer aus. Das Programm bleibt dann erhalten. Drücken Sie **ENTER** oder **=**, während die Asterixe angezeigt werden, um das Programm zu löschen. (Der Computer muß einen Datenbereich und einen Systembereich in der RAM-Karte reservieren können. Sollte dies nicht möglich sein, erscheint die obige Anzeige.)

**2) Verwendung von Programmen, die mit diesem Computer geschrieben wurden, auf anderen Modellen**

Die Programme, die mit dem PC-1450 geschrieben wurden, können auf dem PC-1350 und dem PC-2500 eingesetzt werden.



Führen Sie die folgende Bedienung durch und beachten Sie die folgenden Punkte:

1. Schalten Sie die Spannungsversorgung ein.

2. Drücken Sie CLEAR **ENTER**.

Führen Sie die folgende Bedienung durch und beachten Sie die folgenden Punkte:

1. Schalten Sie die Spannungsversorgung ein.

2. Geben Sie **3** ein, um den BASIC-Modus zu wählen.

3. POKE&6D97,0,&60 **ENTER**

4. CLEAR **ENTER**

- Programme, die Befehle enthalten, die nicht für den PC-1350 und/oder PC-2500 verfügbar sind, müssen geändert werden.
- Vorher eingegebene Daten können nicht verwendet werden und werden gelöscht.

## 1.3 GRUNDLAGEN SHARP CE-126P

### 1.3.1 ALLGEMEINES

Die Option **SHARP CE-126P** (Thermodrucker mit integriertem Kassetten-Interface) ermöglicht es Ihnen, Ihre mit dem **PC-1450** erstellten Programme oder Daten auszudrucken bzw. mit einem externen Kassettenrecorder zu speichern.

Eigenschaften des **CE-126P**:

- 24-Zeichen-Thermodrucker
- Manueller oder programmgesteuerter Betrieb des Druckers
- 300-Baud-Kassetten-Interface
- Manueller oder programmgesteuerter Betrieb des Kassetten-Interface
- Trockenbatterie-Betrieb (Transportfähigkeit)/Netzadapterbetrieb

Der Anschluß des **CE-126P** an Ihren Computer ist in dem, dem **CE-126P** beigelegten Handbuch beschrieben.

### 1.3.2 VERWENDUNG DES DRUCKERS

#### 1.3.2.1 Manueller Druckbetrieb (Protokollerstellung)

Der manuelle Druckbetrieb ermöglicht es Ihnen, Ihre Rechnungen sofort auf Papier zu protokollieren. Dieser Betrieb wird durch Drücken der **[SHIFT]**- und der **[ENTER]**-Taste eingeschaltet. Am Display wird dies durch einen kleinen waagrechten Strich oberhalb von 'PRINT' angezeigt.



Auf dem Thermopapier wird bei der Protokollführung die gleiche Zeichenfolge in derselben Form, wie am Display angezeigt, ausgedruckt.

Nach Abschluß der Eingabe mit **[ENTER]** wird der Druck gestartet. Vorher können Sie Ihre Eingaben, wenn notwendig, noch korrigieren.

Das folgende Bild zeigt eine Protokollführung als Beispiel:

```

12000*.065
780.
780+25.6
805.6
SIN 45
7.071067812E-01
F=50
50.
X=2*PI*F
314.1592654
P=60*X
18849.55592
P/32
589.0486225

```

### 1.3.2.2 Programmgesteuerter Ausdruck

Enthalten BASIC-Programme LPRINT-Anweisungen, so können die den Anweisungen folgenden Werte oder Textausdrücke über den angeschlossenen Thermo drucker (Option **CE-126P**) ausgedruckt werden.

Programme, die mit PRINT-Anweisungen geschrieben wurden, können ganz einfach umgewandelt werden, so daß die Ausgabe nicht über das Display, sondern über den angeschlossenen Drucker erfolgt. Dies wird mit dem Befehl **PRINT = LPRINT** erreicht, der in einer Programmzeile dem Programm vorangestellt oder im RUN-Mode direkt eingegeben werden kann.

Im letzteren Fall muß das Programm jedoch mit **[DEF]**-Taste oder **GOTO < Zeilennummer >** gestartet werden.

Mit dem Befehl **PRINT = PRINT** kann dieser Befehl wieder aufgehoben werden.

Die Umschaltung kann auch in Abhängigkeit von einem logischen Vergleichsausdruck innerhalb einer IF-Anweisung erfolgen.

Das im Hauptspeicher gespeicherte Programm kann mittels der **LLIST**-Anweisung am Drucker gelistet werden. Näheres hierzu siehe **LLIST**-Anweisung, Seite 122.

Ist die zu druckende Programmzeile länger als 24 Zeichen, so erfolgt die Ausgabe am Drucker zwei- bzw. mehrzeilig. Dabei wird der Druck ab der zweiten Zeile um vier bzw. um sechs Stellen eingerückt, um den Ausdruck übersichtlicher zu gestalten.

#### Hinweise:

1. Wenn während des Drucks ein Fehler auftritt, der im Zusammenhang mit nicht richtig transportiertem Papier steht, so reißen Sie das Papier an der Abrißkante ab und ziehen das verbleibende Papier aus dem Drucker.  
Löschen Sie die Fehlermeldung durch Drücken der **[C-C]**-Taste und legen Sie die Papierrolle wieder richtig ein.
2. Wenn der Drucker durch irgendwelche externe elektrische Störfelder beeinflußt wird, kann es sein, daß der Ausdruck fehlerhaft ist. Drücken Sie die **[BRK]**-Taste, um den Druck zu unterbrechen. Schalten Sie den Thermo drucker **CE-126P** aus und nach einigen Sekunden wieder ein. Initialisieren Sie den Drucker durch Drücken von **[C-C]**.
3. Schalten Sie den Drucker nur ein, wenn Sie ihn benützen (Schonung der Batterien).

### 1.3.3 Verwendung des integrierten Kassetten-Interfaces

Das integrierte Kassetten-Interface ermöglicht es Ihnen, Ihre Programme bzw. Daten auf Band zu speichern. Dazu benötigen Sie einen für Datenaufzeichnung geeigneten Kassettenrekorder, z.B. den **CE-152**.

Einmal gespeicherte Programme oder Daten können ganz leicht wieder in den Rechner geladen werden.

#### 1.3.3.1 Auswahl des Kassettenrekorders

Wir empfehlen Ihnen, für Ihre Programmspeicherung den extra dafür vorgesehenen Kassettenrekorder **CE-152** zu verwenden. Der **CE-152** wurde eigens zur Speicherung der mit SHARP-Taschencomputern erstellten Programme oder Daten entwickelt.

Wenn Sie jedoch einen anderen Rekorder verwenden wollen, sollte dieser folgende Ausstattung haben:

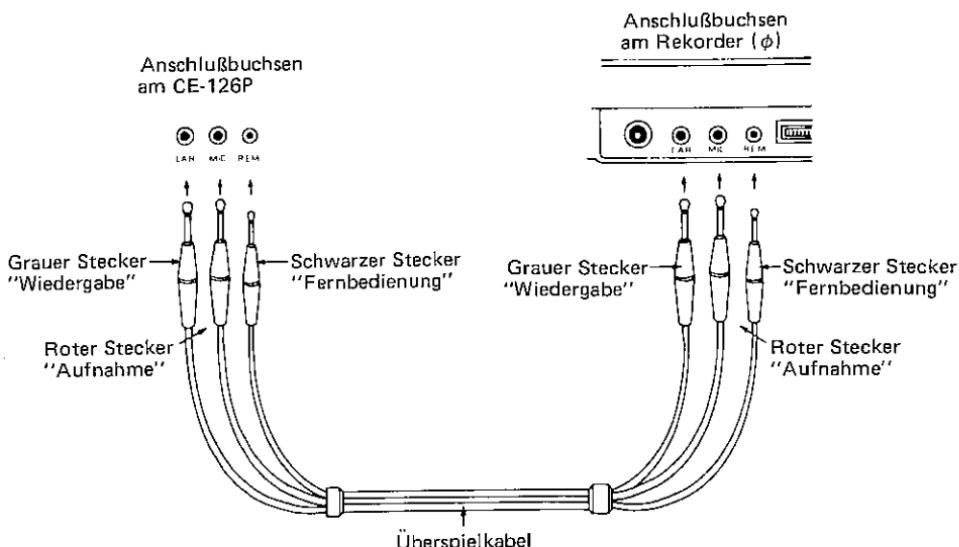
- Fernbedienung (REMOTE), Klinkenbuchse 2,5 mm  $\phi$
- Bandzählwerk
- Mikrofoneingang (MIC), Klinkenbuchse 3,5 mm  $\phi$
- Ausgang (EAR), Klinkenbuchse 3,5 mm  $\phi$

Daneben sind folgende technische Daten gefordert:

- Eingangsimpedanz (MIC)	200...1000 Ohm
- Eingangsempfindlichkeit	Kleiner als 3 mV (-50 dB)
- Ausgangsimpedanz (EAR)	Kleiner als 10 Ohm
- Ausgangsleistung (EAR)	Größer als 1 V <sub>ss</sub>
- Klirrfaktor	Kleiner als 15% im Bereich zwischen 200...4 kHz
- Gleichlaufschwankungen	0,3% max. (W.R.M.S.)

### 1.3.3.2 Anschluß des Kassettenrekorders an das CE-126P

Die herzustellenden Verbindungen sind in der folgenden Abbildung dargestellt.



### 1.3.4 DATENSPEICHERUNG AUF MAGNETBAND

#### 1.3.4.1 Allgemeines

Auf dem Magnetband können Sie Daten und Programme abspeichern oder von Band zurückladen. Außerdem können Sie das Magnetband als externen Programmspeicher verwenden, um große Programme ablaufen zu lassen.

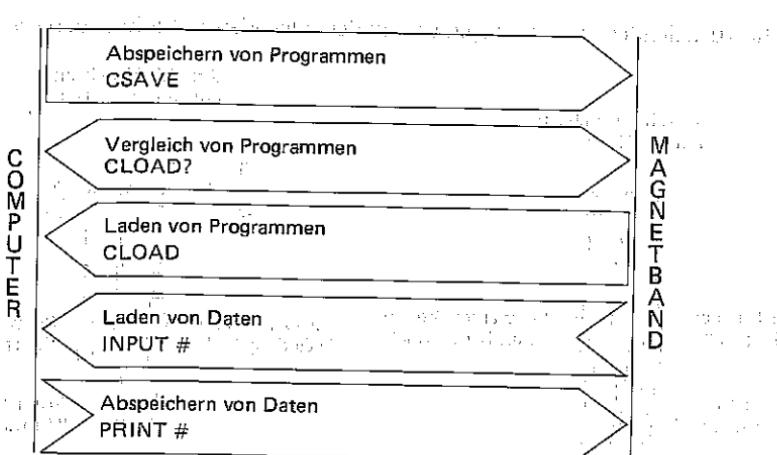
Die Informationen aus Daten und Programmen werden in Blöcken nach dem Zweitonverfahren auf das Magnetband geschrieben.

Auf einem Magnetband können abhängig vom Kassettentyp sehr viele Programme gespeichert werden. Damit Sie die Daten oder Programme wiederfinden und unterscheiden können, beginnt jeder Block aus einem Blocknamen (Programmnamen), der aus maximal 7 Zeichen besteht. Abgeschlossen wird der Speichervorgang, wenn die gesamte Information abgespeichert ist.

Um eine sichere Trennung der Blöcke zu gewährleisten, schreibt der Rechner vor dem Blocknamen automatisch etwa 5 bis 7 Sekunden lang einen konstanten Signaltion. Ein kompletter Block hat damit folgende Form:

Signalton	Blockname	Programme oder Daten	
-----------	-----------	----------------------	--

Folgende Möglichkeiten werden bei der Datenspeicherung auf Magnetband geboten:



#### Hinweis:

Auf die Informationen auf Band können Sie nur sequentiell zugreifen. Damit Sie Ihre Daten und Programme schneller wiederfinden, sollten Sie sich vor dem Abspeichern Informationen über den Zählerstand des Kassettenrekorders und den Blocknamen der Daten aufschreiben. Sie werden sehr schnell eine umfangreiche Programmabibliothek von Daten erhalten.

#### 1.3.4.2 Speichern von Daten und Programmen

Schließen Sie Ihren Kassettenrekorder wie in Abschnitt 1.3.3.2 beschrieben an Ihr CE-126P an. Nachdem Sie Ihr Programm eingegeben haben, können Sie den Rekorder für die Abspeicherung vorbereiten:

- Stellen Sie den REMOTE-Schalter des Interfaces auf OFF.
- Legen Sie eine Kassette in den Rekorder ein.
- Vergewissern Sie sich, daß die eingelagerte Kassette zurückgespult ist.
- Suchen Sie eine freie Stelle am Band.  
Merken Sie sich den Zählerstand.
- Stellen Sie den REMOTE-Schalter des Interfaces auf ON.
- Bereiten Sie die Aufnahme vor:  
“RECORD”- und “PLAY”-Taste drücken. Lautstärkeregler auf Mitte bis Maximum stellen; Tonhöhenregler auf “Höhen”.
- Kommando Eingabe:  
Mit der CSAVE-Anweisung können Sie Ihr Programm gleichzeitig mit einem Namen, dem Blocknamen (Programmnamen) versehen.

Wenn Sie die **ENTER**-Taste betätigt haben, setzt sich das Magnetband in Bewegung. Ihr Programm wird jetzt übertragen und unter dem angegebenen Namen abgespeichert. Zu Beginn hören Sie 5 bis 7 Sekunden den konstanten Signalton, anschließend eine Folge von Tönen. Sobald der Computer am Ende des Programms angelangt ist, stoppt das Band. Das Bereitschaftssymbol wird auf der Anzeige des Computers angezeigt.

### 1.3.4.3 Überprüfen der Abspeicherung

Nach der Abspeicherung Ihres Programms mit der CSAVE-Anweisung und bevor Sie das Programm im Computer löschen, ist es empfehlenswert, zu überprüfen, ob es fehlerfrei übertragen wurde. Das Band könnte beispielsweise eine schadhafte Stelle haben.

Die Überprüfung ist ganz einfach und erfolgt mit der CLOAD?-Anweisung. Der Computer vergleicht das Programm in seinem Speicher mit den auf dem Band gespeicherten Informationen. Ist die Aufzeichnung fehlerfrei, wird nach Abschluß der Überprüfung am Display des Computers wieder das Bereitschaftszeichen angezeigt.

Stellt der Computer beim Vergleich der Informationen einen Fehler fest, erfolgt die Fehlermeldung ERROR 8.

Löschen Sie das aufgezeichnete Programm und versuchen Sie es nochmals zu speichern. Gegebenenfalls verwenden Sie eine andere Bandstelle bzw. eine andere Kassette.

### 1.3.4.4 Laden von Programmen oder Daten

Beim Laden von Programmen gehen Sie wie folgt vor:

- Lautstärkeregler auf Mitte bis Maximum stellen; Tonhöhenregler auf "Höhen". Sollte der Kassettenrekorder bei Einstellung des Lautstärkereglers auf Maximum nicht ordnungsgemäß arbeiten, verringern Sie die Lautstärke und probieren erneut.
- Spulen Sie das Band an den Anfang zurück.
- Stellen Sie das Zählwerk auf Null.
- Suchen Sie mit Hilfe des Zählwerks die Bandstelle, an der Ihr Programm beginnt.
- Schalten Sie den REMOTE-Schalter des Kassetten-Interface auf ON.
- Drücken Sie die "PLAY"-Taste Ihres Kassettenrekorders.
- Geben Sie ein:

CLOAD "PROG. 1"

Wenn Sie die **ENTER**-Taste gedrückt haben, setzt sich das Magnetband in Bewegung. Sie hören die gleiche Signalfolge wie bei der Abspeicherung. Am Display wird während des Ladevorgangs ein Asterix (\*) angezeigt.

Bei der Eingabe der CLOAD-Anweisung gibt es zwei Möglichkeiten: **CLOAD "Programmname"** oder **CLOAD "Programmname"**.

Dabei wird das Programm vom Magnetband in den Speicher des Rechners übertragen. Die beiden CLOAD-Anweisungen unterscheiden sich nur dadurch, daß bei der zweiten Anweisung der Blockname (Programmname) zusätzlich mit eingegeben wird. Damit wird erreicht, daß nur jenes Programm, dessen Blockname mit dem eingegebenen übereinstimmt, übertragen wird. Alle anderen auf Band gespeicherten Programme werden überlesen.

**1.4 SERIELLE E/A-FUNKTIONEN**

Der Computer ist mit einem seriellen E/A-Interface ausgerüstet. Hier kann der Computer zum Beispiel an einen Personalcomputer angeschlossen werden, um Daten auszutauschen.

Achtung: Vor dem Anschluß sollten die Spannungswerte überprüft werden, da eine Überspannung interne Bauteile des Computers zerstören könnte.

## ALLGEMEINE HANDHABUNG DES SERIELLEN E/A-INTERFACE

Normalerweise ist das serielle Interface geschlossen. Dann können natürlich weder Daten empfangen noch Daten gesendet werden. Deshalb ist der erste Schritt des Datenaustausches das Öffnen des Kanals mit dem OPEN-Kommando. Sollte der Kanal allerdings schon geöffnet sein, so meldet der Rechner bei Ausführung des OPEN-Kommandos "ERROR 8".

Weiterhin ist es nötig, die Bedingungen für den Datentransfer festzulegen. Anders ausgedrückt: Die Form der Daten muß sowohl dem sendenden Computer als auch dem angeschlossenen Computer bekannt sein. Erwartet der empfangende Computer die Daten in einer anderen Form als der sendende sie herausgeschickt hat, so werden die Daten verfälscht. Die Bedingungen werden im OPEN-Kommando festgelegt.

Nachdem nun also der Kanal geöffnet ist und sowohl der sendende als auch empfangende Computer die Form der Daten kennt, können Daten mit den folgenden Kommandos ausgetauscht werden:

LPRINT, LLIST, SAVE, LOAD, PRINT #1, INPUT #1

Nach dem Ende des E/A-Programms wird der Kanal automatisch geschlossen. Auch das RUN-Kommando schließt den E/A-Kanal. In einem Programm kann man den Kanal mit dem CLOSE-Kommando schließen.

In einem Programm, das das serielle E/A-Interface benutzt, muß der Kanal also erst geöffnet werden, dann kann man Daten übertragen, und dann sollte der Kanal wieder geschlossen werden, alles wie oben beschrieben.

Achtung: Das Interface des Computers wartet nicht, ob angeschlossene Geräte zum Empfang bereit sind und hat auch keine Zeitfunktion, die den Übertragungsprozeß bei nicht betriebsbereiter Peripherie abschaltet. Die Datenübertragung wird also auch bei zum Beispiel abgeschaltetem Empfänger versucht, da der Computer die Ausführung nicht selbständig unterbrechen kann. In diesem Fall drücken Sie die **BRK**-Taste, um den Rechner zu stoppen.

### Serielle E/A-Funktion bei Verwendung des CE-140P

Der Anschluß des Computers an den Farb-Punktmatrixdrucker CE-140P (Option) ermöglicht Ausdrucken an jedem Ort mit einer einzigen, kompakten Einheit.

An der Seite des Druckers CE-140P befindet sich eine serielle E/A-Schnittstelle zum Anschließen der Computer/Drucker-Einheit an einen Personal Computer.

- \* Der CE-140P Schalter sollte normalerweise auf SIO stehen. Abhängig von der Schalterstellung kann der Drucker wie folgt eingesetzt werden.

Schalterstellung des CE-140P	Serielle E/A-Schnitt- stelle	nur CE-140P ist ange- schlossen	CE-140P und CE-126P sind angeschlossen*1
SIO (Normalerweise auf diese Posi- tion stellen.)	OFFEN	Eingabe/Ausgabe von CE-140P E/A-Schnitt- stelle	Gleich wie links
	GESCHLOSSEN	Ausgabe, gedruckt auf CE-140P	Ausgabe, gedruckt auf CE-126P
PRINTER	OFFEN*2	Ausgabe, gedruckt auf CE-140P	Gleich wie links
	GESCHLOSSEN	Ausgabe, gedruckt auf CE-140P	Ausgabe, gedruckt auf CE-126P

- \*1. Der Thermodrucker CE-126P (Option) wird an den Anschluß an der linken Seite des Computers zusammen mit dem Drucker CE-140P angeschlossen.
  - \*2 Zur Verwendung dieser Konfiguration die Bedienungen für die serielle Eingabe/Ausgabe mit dem OPEN-Kommando spezifizieren, wie unten beschrieben. (Einstellung bei der Initialisierung; siehe unter OPEN-Kommando.)

OPEN "1200, N, 8, 1, A, C, &1A"

Wenn die E/A-Einstellung nicht korrekt ist, arbeitet der Drucker CE-140P nicht ordnungsgemäß.

<sup>10</sup> See, *for example*, the recent discussion by John C. Scott, *What Is a Corporation?* (Berkeley, 1975).

**TEIL II  
PRAXIS**

## **TEIL II PRAXIS**

### **2.1 SHARP PC-1450 – EINSATZ ALS WISSENSCHAFTLICHER RECHNER**

#### **2.1.1 BETRIEBSHINWEISE**

Die folgenden Absätze enthalten einige wichtige Hinweise, die sich auf die CAL-Betriebsart des Rechners beziehen. Die in TEIL I, Abschnitt 1.1 enthaltenen allgemeinen Betriebshinweise haben darüber hinaus Gültigkeit.

Das Umschalten des Rechners in die CAL-Betriebsart (**CAL**-Taste drücken) sowie das für diese Betriebsart relevante Tastenfeld wurden bereits in TEIL I, Kapitel 1.3 aufgezeigt, so daß an dieser Stelle auf eine nähere Erläuterung dieser Punkte verzichtet werden kann.

##### **2.1.1.1 Zweite Funktionsebene der Tastatur; Merkmale; Editier- und Korrekturmöglichkeiten**

Wie aus der nachfolgenden Abbildung ersichtlich, weisen nahezu alle Tasten des Tastenfelds der CAL-Betriebsart eine zweite und, wenn man die Tastenfunktionen für die statistischen Berechnungen (STAT) und die Umwandlung "Hexadezimal-Dezimal" hinzuzählt, eine dritte und vierte Funktionsebene auf.

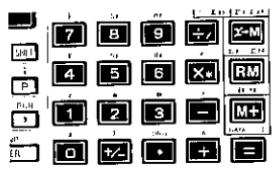
In den Erläuterungen und Rechenbeispielen ist die Auswahl der zweiten Tastenfunktion aus der Verwendung der **SHIFT**-Taste ersichtlich. Die Tastenfunktionen für die statistischen Funktionen und die Umwandlung "Hexadezimal-Dezimal" werden automatisch, d.h. durch das Drücken der **SHIFT STAT**-Taste bzw. **HEX**-Taste vorgewählt.

Bei näherer Betrachtung des Tastenfelds der CAL-Betriebsart wird bereits ein wichtiges Merkmal des **SHARP PC-1450** deutlich. Es steht eine große Anzahl von Tastenfunktionen zur Verfügung, deren Verwendung zu einer wesentlichen Arbeits erleichterung und Arbeitsvereinfachung beiträgt.

Die Tastenfunktionen sind nachfolgend kurz aufgeführt und die entsprechenden Tasten angegeben. In den folgenden Abschnitten sind die einzelnen Funktionen zudem anhand von Berechnungsbeispielen erläutert. Die in Klammern stehende Seitenzahl gibt an, auf welcher Seite das Berechnungsbeispiel zu finden ist.

## Grundrechenfunktionen (Seite 63):

Addition, Subtraktion, Multiplikation Division (Vorzeichenumkehr)



## Speicherrechnung (Seite 65):

Übertragen in den Speicher

**X↔M**

Anzeigen des Speicherinhalts

**RM**

Addition und Subtraktion zum/vom Speicher

**M+**

## Trigonometrische und Hyperbel-Funktionen (Seite 69):

<b>arc hyp</b>	<b>sin<sup>-1</sup></b>	<b>cos<sup>-1</sup></b>	<b>tan<sup>-1</sup></b>
<b>hyp</b>	<b>sin</b>	<b>cos</b>	<b>tan</b>

Trigonometrische und inverse trigonometrische Funktionen (Arcusfunktionen)

<b>sin</b>	<b>cos</b>	<b>tan</b>
<b>SHIFT</b>	<b>arc hyp</b>	<b>sin<sup>-1</sup></b>
		<b>cos<sup>-1</sup></b>
		<b>tan<sup>-1</sup></b>

Hyperbelfunktion und inverse Hyperbelfunktion (Arefunktion)

<b>hyp</b>	<b>sin</b>	<b>cos</b>	<b>tan</b>
<b>SHIFT</b>	<b>archyp</b>	<b>sin<sup>-1</sup></b>	<b>cos<sup>-1</sup></b>
			<b>tan<sup>-1</sup></b>

Umwandlung "Fließkomma-Wissenschaftliche" Schreibweise (Notation z.B. 1000 =  $1 \times 10^3$ ) und Festlegung der Nachkommastellen bzw. der automatischen Aufrundung:**TAB**  
**F+E**

Umwandlung "Fließkomma-Wissenschaftliche" Schreibweise (Seite 57)

**F+E**

Festlegung Nachkommastellen/autom. Rundung (Seite 56)

**SHIFT**    **TAB** (Eingabe Nachkommastellen)

Löschen der letzten Eingabe bzw. einer Fehlermeldung (Seite 55):

**CA** Löschen der letzten Eingabe  
**C-CE** Löschen der letzten Eingabe und des Displayinhalts

Löschen der vorangegangenen Eingabe bzw. des Displayinhalts durch Drücken der roten **C-CE**-Taste. Bei zweimaligem Drücken wird alles außer Speicher "M" gelöscht. Löschen von Speicher "M" durch Drücken von **C-CE** und **F-M**. Die Funktion "CA" ist in Betriebsart "CAL" unwirksam.

Umwandlung Dezimal-Hexadezimal und Hexadezimal-Decimale (Seite 59):

Dezimal-Hexadezimal

**DEC**  
**HEX**

Hexadezimal-Decimale

**HEX**

**HEX** (Eingabe) **SHIFT** **DEC**

Umwandlung von Winkel/Zeit (Seite 58):

**D.MS**  
**DEG**

Grad °/Minuten ' /Sekunden " in Dezimalzahl (Sexagesimal-Decimale)  
 (Std./Min./Sek. in Dezimalzahl)

**DEG**

Dezimalzahl in Grad °/Minuten ' /Sekunden " (Dezimal-Sexagesimal)  
 (Dezimalzahl in Std./Min./Sek.)

**SHIFT** **D.MS**

Natürlicher Logarithmus und Exponentialfunktion der Zahl "e", Hexadezimaleingabe:

**ex** **e**  
**In**

Natürlicher Logarithmus (Seite 70)

**In**

Exponentialfunktion der Zahl "e" (Seite 70)

**SHIFT** **ex**

Hexadezimal."E"

**HEX** **E**

Allgemeiner Logarithmus und Exponentialfunktion  $10^x$ , Hexadezimaleingabe:

**10<sup>x</sup>** **F**

**log**

Allgemeiner Logarithmus (Seite 70)

**log**

Exponentialfunktion  $10^x$  (Seite 70)**SHIFT**       **$10^x$** 

Hexadezimal "F"

**HEX**      **F**

Reziprok-Rechnen und Umwandlung von rechtwinkligen Koordinaten in Polarkoordinaten:

 **$\frac{r}{x}$** 

Reziprok-Rechnen (Seite 65)

 **$\frac{1}{x}$** 

Umwandlung von rechtwinkligen Koordinaten in Polarkoordinaten (Seite 59)

**SHIFT**       **$r\theta$** 

Austausch der im Rechner gespeicherten Zahl (Ersteingabe, Ergebnis) mit der angezeigten Zahl und Anwahl der Betriebsart "Statistische Berechnungen":

**↔**

Austausch

**↔**

Anwahl der Betriebsart "Statistische Berechnungen" (Seite 71)

**SHIFT**      **STAT**

Exponenteneingabe (wissenschaftliche Schreibweise) und Eingabe der Kreiszahl (Pi), Hexadezimaleingabe:

 **$\pi_A$** 

Exponenteneingabe (Seite 57)

**EXP**

Kreiszahl (Pi)

**SHIFT**       **$\pi$** 

Hexadezimal "A"

**HEX**      **A**

Potenzieren und Berechnung der x-ten Wurzel von y, Hexadezimaleingabe (Seite 65):

 **$y^{\sqrt[x]{B}}$** 

Potenzieren

 **$y^x$** 

Wurzelziehen

**SHIFT**       **$\sqrt[x]{y}$** 

Hexadezimal "B"

**HEX**      **B**

## Berechnung von Quadrat- und Kubikwurzel, Hexadezimaleingabe (Seite 64):

Quadratwurzel

Kubikwurzel

Hexadezimal "C"

## Quadrierung und Prozentberechnung, Hexadezimaleingabe (Seite 65):

Quadrierung der angezeigten Zahl

Prozentberechnung (Prozentsatz, prozentuale Zunahme)

Hexadezimal "D"

## Öffnen der Klammer und Umwandlung von Polarkoordinaten in rechtwinklige Koordinaten, statistische Berechnungen:

Öffnen der Klammer

## Umwandlung von Polarkoordinaten in rechtwinklige Koordinaten (Seite 59)

## Statistische Berechnungen (Seite 71)

Datenanzahl

Datensumme von xy

Schließen der Klammer und Fakultätswert, statistische Berechnungen:

Schließen der Klammer

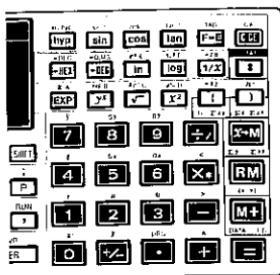
Fakultätswert (Seite 70)

## Statistische Berechnungen (Seite 71)

Datensumme von x

"STAT"  $\Sigma x$ Summe von  $x^2$ "STAT" **SHIFT**  $\Sigma x^2$ 

## Statistische Berechnungen (Seite 71):



## Anwahl der Betriebsart "Statistische Berechnungen"

**SHIFT** **STAT**

Operandenanzahl

 $n$ 

Datensumme von xy

**SHIFT**  $\Sigma xy$ 

Datensumme von x

 $\Sigma x$ Summe von  $x^2$ **SHIFT**  $\Sigma x^2$ 

Mittelwert der Daten y

**SHIFT**  $\bar{y}$ 

Standardabweichung der y-Werte mit Gesamtheitsparameter "n-1"

**SHIFT**  $S_y$ 

Standardabweichung der y-Werte mit Gesamtheitsparameter "n"

**SHIFT**  $\sigma_y$ 

Datensumme von y

 $\Sigma y$ Summe von  $y^2$ **SHIFT**  $\Sigma y^2$ 

Mittelwert der Daten x

**SHIFT**  $\bar{x}$

Standartabweichung der x-Werte mit Gesamtheitsparameter "/n-1"

**SHIFT** **Sx**

Standartabweichung der x-Werte mit Gesamtheitsparameter "n"

**SHIFT** **σx**

Eingabe von x- und y-Daten

**(x,y)**

Korrelationskoeffizient

**SHIFT** **r**

Schnittpunkt der Regressionslinie mit der y-Achse

**SHIFT** **a**

Steilheit der Regressionslinie

**SHIFT** **b**

Dateneingabe

**DATA**

Korrektur einer falschen Eingabe

**SHIFT** **CD**

Näherungswert von x bei vorgegebenem y-Wert

Eingabe y-Wert **SHIFT** **x'**

Näherungswert von y bei vorgegebenem x-Wert

Eingabe x-Wert **SHIFT** **y'**

Anwahl der Winkeleinheit (Seite 54):

DEG (Grad; °, ', ")

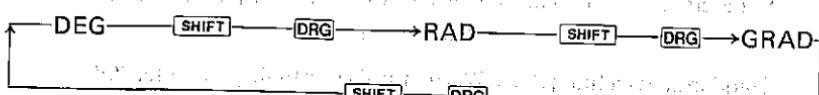
**SHIFT** **DRG**

RAD (Radian; rad)

**SHIFT** **DRG**

GRAD (Neugrad; gon)

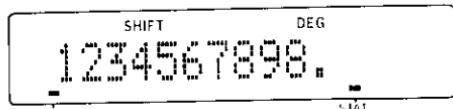
**SHIFT** **DRG**



Anmerkung:

Die Funktionen (CA), **<**, **>** und **↑** werden in der Betriebsart "CAL" nicht verwendet.

### 2.1.1.2 Anzeige/Display-Format und Symbole.



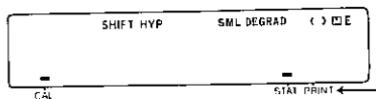
Fließkommaschreibweise.  
(Normalformat)



Wissenschaftliche  
Schreibweise

Mantisse (12 Stellen) Exponent (4 Stellen)

Die Anzeige des PC-1450 umfaßt 16 Stellen. In der Betriebsart "CAL" werden die Rechenergebnisse soweit möglich in Fließkommaschreibweise dargestellt. Ist das Rechenergebnis kleiner 0,000000001 oder größer 9999999999 (größer -0,000000001 oder kleiner -9999999999), erfolgt die Anzeige in wissenschaftlicher Schreibweise. Dabei wird die Mantisse 12stellig (einschließlich Vorzeichen und Komma) und der Exponent 4stellig (einschließlich Symbol und Vorzeichen) angezeigt.



Ausdruck über Drucker  
siehe 1.3 "GRUNDLAGEN  
SHARP CE-126P"

Die Betriebsart "CAL" verwendet die oben abgebildeten Symbole und Markierungen, die nachfolgend aufgelistet sind und deren Bedeutung erläutert wird.

- CAL** Die Markierung (—) über der Beschriftung "CAL" gibt an, daß sich der Rechner in der Betriebsart "CAL" befindet.
- SHIFT** Das Wort "SHIFT" wird angezeigt, wenn die [SHIFT]-Taste gedrückt wurde und damit die zweite Funktionsebene der Tasten angewählt ist.
- HYP** Das Wort "HYP" wird angezeigt, wenn die [hyp]-Taste gedrückt wurde und damit eine Hyperbelfunktion angewählt ist. Werden die Tasten [SHIFT] und [hyp] gedrückt (Anzeige "SHIFT HYP"), so ist damit eine inverse Hyperbelfunktion (Arefunktion) angewählt.
- SML** Die Taste [SML] wird für die Ausgabe von Kleinbuchstaben benutzt. Wenn Sie die Taste [SML] erneut drücken, werden wieder Großbuchstaben angezeigt.

**DEG** Diese drei Wörter können durch wiederholtes Betätigen der Tasten **SHIFT** und **DRG** angewählt werden. Die Wörter geben an, in welcher Winkeleinheit gerechnet wird.

DEG : (Grad; °, ', ")

RAD : (Radian; rad)

GRAD : (Neugrad; gon)

$$\text{Gestreckter Winkel} = 180^\circ = \pi \text{ rad} = 200 \text{ gon}$$

( ) Das Klammersymbol erscheint, sobald beim Eintippen einer Berechnungsformel die **[ ]**-Taste verwendet wird.

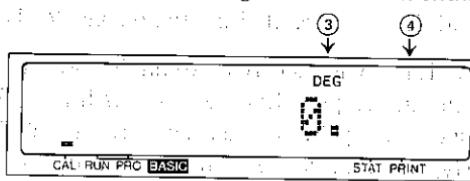
**M** Dieses Symbol zeigt an, daß sich ein numerischer Wert, der ungleich Null ist, im Speicher befindet und der Speicher somit belegt ist.

**E** Das Symbol "E" in der rechten oberen Ecke des Anzeigefelds zeigt an, daß eine Bereichsüberschreitung aufgetreten ist oder eine falsche Anweisung gegeben wurde. Der Fehlerzustand kann durch das Betätigen der **C-CE**-Taste beseitigt werden.

**STAT** Die Markierung (-) über der Beschriftung "STAT" im rechten unteren Bereich des Anzeigefelds gibt an, daß sich der Rechner in der Betriebsart "Statistische Berechnungen" befindet. Die Betriebsart wird durch Drücken der Tasten **SHIFT** und **STAT** angewählt und durch nochmaliges Drücken wieder ausgeschaltet.

### 2.1.1.3 Grundeinstellung

Bevor mit den Berechnungen begonnen wird, sollte die Grundeinstellung des Rechners überprüft werden. Dazu den POWER-Schalter auf ON schalten oder, falls der Rechner automatisch abgeschaltet hat, die **ON** **BRK**-Taste drücken. Anschließend die **CAL**-Taste (grün) drücken, um so die Betriebsart "CAL" anzuwählen und zweimal die **C-CE**-Taste betätigen. Das Anzeigefeld sollte nun, entsprechend der nachfolgenden Abbildung, nur die Anfangsinformationen enthalten.



Falls dies nicht der Fall ist, die folgenden Kurzanweisungen lesen und die erforderlichen Maßnahmen durchführen.

- ① Es wird mehr als eine Null angezeigt (z.B. 0.000):

Die Anzahl der Nachkommastellen ist festgelegt. Durch Drücken der Tasten **SHIFT** **TAB** **[ ]** auf Fließkomma mit vollem Anzeigeformat umschalten.

- ② Über der Beschriftung "STAT" wird eine Markierung (-) angezeigt:  
Die Betriebsart "Statistische Berechnungen" ist angewählt.  
Durch Drücken der Tasten **SHIFT STAT** die Betriebsart aufheben.
- ③ Statt DEG wird RAD oder GRAD angezeigt:  
Die Winkeleinheit ist in Radian (rad) oder Neugrad (gon) vorgewählt.  
Durch Drücken der Tasten **SHIFT DRG** Winkeleinheit auf DEG einstellen.
- ④ Das Symbol **M** wird angezeigt:  
Der Speicher ist mit einem numerischen Wert belegt. Löschen des Speichers durch Drücken der Tasten **C-CE X+M**.
- ⑤ Alle im oberen Bereich des Anzeigefelds dargestellten Symbole, mit Ausnahme der in Punkt ③ und ④ beschriebenen, können durch Drücken der **C-CE**-Taste gelöscht werden.

#### 2.1.1.4 Löschen der Eingabe oder einer Fehlermeldung

Durch einmaliges Drücken der roten **C-CE**-Taste wird die letzte Eingabe bzw. der Displayinhalt gelöscht. Ein zweimaliges Drücken der **C-CE**-Taste löscht auch den zuerst eingegebenen Wert bzw. das Ergebnis der vorangegangenen Berechnung.

Der Speicherinhalt **M** wird dadurch nicht gelöscht.

123 **+** 456 4 5 6 .

**C-CE** Ø .

789 **=** 9 1 2 .

(123 + 789 = 912)

6 **X** 2 **+** 1 2 .

**C-CE** Ø .

6 **÷** 2 **+** 3 .

5 **=** 8 .

Die **C-CE**-Taste dient darüber hinaus zum Löschen der Fehlermeldung (E).

5 **÷** 0 **=** Ø . E ← Fehler (Error)-Symbol

**C-CE** Ø .

### 2.1.1.5 Auf-/Abrundung bzw. Festlegung der Nachkommastellen

Die Auf- oder Abrundung des Rechenergebnisses erfolgt in der Regel in der 11. Stelle. Da das Display 10 Stellen aufweist, ist die letzte Stelle des angezeigten Rechenergebnisses nicht auf- oder abgerundet. Bei Benützung der Tasten [=], [**M+**] oder [**A%**] wird der angezeigte, nicht gerundete Wert im Falle weiterer Berechnungen verwendet.

Beispiel:

$$\begin{array}{r} 1 \quad \div \quad 3 \quad \times \quad 3 \quad = \quad 1. \\ 1 \quad \div \quad 3 \quad = \quad 0.333333333 \\ \times \quad 3 \quad = \quad 0.999999999 \end{array}$$

Die Anzahl der Nachkommastellen und damit die Rundung kann frei festgelegt werden. Dazu dient die **TAB**-Taste in Verbindung mit den Tasten (0) bis (9) und (.). Die Festlegung der Nachkommastellen wird durch Drücken der Tasten **SHIFT TAB** und **•** aufgehoben.

- SHIFT TAB 0** → Keine Nachkommastellen. Entsprechend der 1. Nachkommastelle wird auf- oder abgerundet.
- SHIFT TAB 1** → Eine Nachkommastelle. Entsprechend der 2. Nachkommastelle wird auf- oder abgerundet.
- SHIFT TAB 9** → Neun Nachkommastellen. Entsprechend der 10. Nachkommastelle wird auf- oder abgerundet.
- SHIFT TAB •** → Rückstellung auf Fließkomma. Entsprechend der 12. Stelle bzw. der 10. Nachkommastelle bei wissenschaftlicher Schreibweise wird auf- oder abgerundet.

Beispiel:

- |                    |   |
|--------------------|---|
| <b>SHIFT TAB 9</b> | → 0.055555556   |
| <b>• 5 ÷ 9 =</b>   | Neun Nachkommastellen. Die 9. Nachkommastelle wurde aufgerundet.              |
| <b>F+E</b>         | → 5.555555556E-02   |
|                    | Neun Nachkommastellen. Die 9. Nachkommastelle der Mantisse wurde aufgerundet. |
| <b>SHIFT TAB 3</b> | → 5.556E-02   |
|                    | Drei Nachkommastellen. Die 3. Nachkommastelle der Mantisse wurde aufgerundet. |

**F<sub>4</sub>E**

→ 0.056

Drei Nachkommastellen. Die 3. Nachkommastelle wurde aufgerundet.

**SHIFT TAB**

→ 0.0555555555

Der Wert ist im Rechner in der Form von  $5.55555555555 \times 10^{-2}$  gespeichert.

Die Rundung in der 9. Nachkommastelle der Mantisse ergibt 5.555555556  $\times 10^{-2}$ .

Da das Display 10 Stellen anzeigt, ist die Aufrundung in der Ergebnisanzeige nicht enthalten.

**F<sub>4</sub>E**

→ 5.555555556E-02

### 2.1.1.6 Wissenschaftliche Schreibweise

Die Displayanzeige erfolgt, solange die Anzeigekapazität nicht unter- oder überschritten wird, in Festkommaschreibweise. Durch Drücken der **F<sub>4</sub>E**-Taste kann jedoch auf einfache Weise zwischen "Festkomma-" und "Wissenschaftlicher Schreibweise" gewählt werden.

Soll oder muß ein Wert in wissenschaftlicher Schreibweise eingegeben werden, ist zur Eingabe des Exponenten die **EXP**-Taste zu verwenden. Das Vorzeichen des Exponenten läßt sich mit der **+/-**-Taste bestimmen.

Beispiel: 23 **X** 1000 **=** 23000.

**F<sub>4</sub>E**2.3E 04**F<sub>4</sub>E**23000.4 **π A EXP** 34.E 03(4 x 10<sup>3</sup>)**=**4000.1.23 **+/-**-1.23**EXP** 5 **+/-**-1.23E-05(-1.23 x 10<sup>-5</sup>)**=**-0.0000123

### 2.1.1.7 Umwandlung des Winkels und der Zeit

Zur Umwandlung von Winkel- oder Zeitangaben ( $^{\circ}$ ,  $'$ ,  $"$  bzw. Std., Min., Sek.) in den entsprechenden Dezimalwert muß der Wert als ganze Zahl ( $^{\circ}$ ; Std.) plus Nachkommastellen ( $'$ ,  $"$ ; Min., Sek.; jeweils zweistellig) eingegeben werden.

Aufgabe: Umwandlung von  $12^{\circ} 47' 52''$  in den entsprechenden Dezimalwert

Tastenfolge: 12.4752 **DEG**

Ergebnis: 12.7977778

Bei der Umwandlung des dezimalen Winkel-/Zeitwerts in den sexagesimalen Wert ( $^{\circ}$ ,  $'$ ,  $"$  bzw. Std., Min., Sek.) ist das angezeigte Ergebnis wie folgt zuzuordnen:

Ganze Zahl	Nachkommastellen		
	1 und 2	3 und 4	5 bis 9
$^{\circ}$ (Std.)	$'$ (Min.)	$"$ (Sek.)	Zehntelsekunden

Aufgabe: Umwandlung des dezimalen Winkelwerts "24,7256" in den entsprechenden Sexagesimalwert

Tastenfolge: 24.7256 **SHIFT DMS**

Ergebnis: 24.433216     $24^{\circ} 43' 32''$

Beispiel:

Ein Pferd läuft Rundenzeiten von 2 Min. 25 Sek., 2 Min. 38 Sek. und 2 Min. 22 Sek. Wie hoch ist die durchschnittliche Rundenzeit?

Tastenfolge: .0225 **DEG** + .0238 **DEG** + .0222 **DEG** =

Zwischenergebnis: 0.123611111

Tastenfolge: **÷** 3 **=**

Zwischenergebnis: 0.041203703

Tastenfolge: **SHIFT DMS**

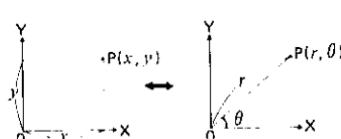
Ergebnis: 0.022833333

Die durchschnittliche Rundenzeit ist

2 Minuten 28 Sekunden

### 2.1.1.8 Koordinatenumwandlung

Umwandlung von rechtwinkligen Koordinaten in Polarkoordinaten ( $x, y \rightarrow r, \theta$ )



$$r = \sqrt{x^2 + y^2} \quad \text{DEG: } 0 \leq |\theta| \leq 180$$

$$\theta = \tan^{-1} \frac{y}{x} \quad \text{RAD: } 0 \leq |\theta| \leq \pi$$

$$\theta = \tan^{-1} \frac{y}{x} \quad \text{GRAD: } 0 \leq |\theta| \leq 200$$

Aufgabe: Umwandlung der rechtwinkligen Koordinaten  $x = 6$  und  $y = 4$  in Polarkoordinaten. Winkeleinheit DEG (°).

Tastenfolge: 6  $\downarrow$  4 SHIFT  $\leftrightarrow\theta$

Ergebnis: 7.211102551 (r)

Tastenfolge:  $\downarrow$

Ergebnis: 33.69006753 ( $\theta$ )

Aufgabe: Umwandlung des Real- und Imaginärteils eines Vektors in Betrag und Richtung (Phase). Winkeleinheit DEG (°).

Realteil  $x = 12$

Imaginärteil  $y = j9$

Tastenfolge: 12  $\downarrow$  9 SHIFT  $\leftrightarrow\theta$

Ergebnis: Betrag  $r = 15$

Tastenfolge:  $\downarrow$

Ergebnis: Richtung (Phase)  $\theta = 36.86989765$

Umwandlung von Polarkoordinaten in rechtwinklige Koordinaten (Formel)

Aufgabe: Umwandlung der Polarkoordinaten  $r = 14$  und  $\theta = \pi/3$  in rechtwinklige Koordinaten. Winkeleinheit RAD (rad).

Tastenfolge: SHIFT DRG

SHIFT  $\pi$   $\div$  3 =  $\downarrow$  14  $\downarrow$  SHIFT  $\leftrightarrow xy$

Ergebnis: 7.000000002 (x)

Tastenfolge:  $\downarrow$

Ergebnis: 12.12435565 (y)

Anmerkung: Es wird zuerst  $\pi/3$  errechnet. Nach der Eingabe von  $r = 14$  werden die beiden Werte durch Drücken der  $\downarrow$ -Taste ausgetauscht und die richtige Reihenfolge so wieder hergestellt.

### 2.1.1.9 Umwandlung "Hexadezimal – Dezimal" und Rechnen mit Zahlen in Hexadezimal-Schreibweise

Die einfache Umwandlung von Zahlen zur Basis 10 (Dezimal) in Zahlen zur Basis 16 (Hexadezimal) und umgekehrt ist auf dem Feld der Datenverarbeitung und

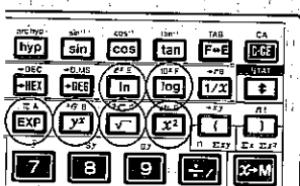
Computertechnik von großer Bedeutung, da in diesen Bereichen die Hexadezimal-Schreibweise häufig anzutreffen ist.

Die Hexadezimal-Schreibweise verwendet die Zahlen 0 bis 9 sowie die Buchstaben A bis F, wobei diese in Dezimal-Schreibweise den Werten 0 bis 15 entsprechen:

Dezimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadezimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Zur Eingabe einer Zahl in Hexadezimal-Schreibweise muß zuerst die **[HEX]**-Taste gedrückt werden. Die Anzeige "HEX" weist darauf hin, daß es sich bei dem angezeigten Wert um eine Sedenzimalzahl (Zahl in Hexadezimal-Schreibweise) handelt.

Die Buchstaben A bis F können nach Drücken der **[HEX]**-Taste über die in der nebenstehenden Abbildung hervorgehobenen Tasten eingegeben werden.



Das Umschalten auf Dezimal-Schreibweise erfolgt durch Drücken der Tasten **[DEC]**.

### Umwandlung "Dezimal-Hexadezimal"

Aufgabe: Umwandlung von Dezimal 30 in Hexadezimal

Tastenfolge: 30 **[HEX]**

Anzeige: **1 E. HEX**

Anmerkung: Zur Eingabe einer weiteren Dezimalzahl durch Drücken der Tasten **[SHIFT]** + **[DEC]** wieder auf Dezimal-Schreibweise umschalten.

Aufgabe: Umwandlung von Dezimal -2 in Hexadezimal

Tastenfolge: **2** **[+/-]** **[HEX]**

Anzeige: **FFFFFFFFFFE. HEX**

Anmerkung: Die Umwandlung einer negativen Dezimalzahl erfolgt intern über das "Zweierkomplement". Das Resultat wird im "16er-Komplement" angezeigt.

Mit der **[+/-]**-Taste kann das Vorzeichen der angezeigten Zahl gewechselt werden. Bei Hexadezimal-Schreibweise erfolgt die Anzeige des jeweiligen 16er-Komplements.

Aufgabe: Umwandlung von Dezimal 123,4 in Hexadezimal

Tastenfolge: **[SHIFT]** **[DEC]** 123,4 **[HEX]**

Anzeige: **7B. HEX**

Anmerkung: Es können nur ganze Zahlen in Hexadezimal umgewandelt werden.  
Die Nachkommastellen bleiben bei der Umwandlung unberücksichtigt.

### Umwandlung "Hexadezimal-Dezimal"

Aufgabe: Umwandlung von Hexadezimal 2BC in Dezimal

Tastenfolge: C-CE HEX 2 B C SHIFT DEC

Anzeige: 700.

Aufgabe: Umwandlung von Hexadezimal FFFFFFFF12 in Dezimal

Tastenfolge: C-CE HEX FFFFFFFF 12 SHIFT DEC

Anzeige: FFFFFFFF12. HEX

-238.

Anmerkung: Die Umwandlung von Hexadezimal FDABF41C01 bis FFFFFFFFFF ergibt eine negative Dezimalzahl.

### Rechnen mit Zahlen in Hexadezimal-Schreibweise

In der Betriebsart "HEX" können auch Rechenoperationen durchgeführt werden.

Dabei sind folgende Hinweise zu beachten:

- Es kann nur mit ganzen Zahlen gerechnet werden, die -Taste ist daher ohne Funktion.
- Bei Divisionen wird immer eine ganze Zahl als Ergebnis angezeigt. Etwaige Nachkommastellen bleiben unberücksichtigt.
- Beispiel: B ÷ 3 = ... 3. HEX
- Falls mit einem Zwischenergebnis weitergerechnet wird, das Nachkommastellen aufweist, erfolgt eine Fehlermeldung (Symbol "E" wird angezeigt).
- Beispiel: B ÷ 3 X ... Error (Symbol "E")
- Bei Kettenrechnungen sind ggf. Klammern zu verwenden, da keine Beachtung der Vorrangordnung erfolgt.
- Durch Drücken der +/- -Taste kann auf einfache Weise das Komplement der angezeigten Sedenzimalzahl ermittelt werden.

Beispiel: A B +/- → FFFFFFFF55. HEX  
+/- → AB. HEX

Aufgabe: A4 + BA

Tastenfolge: A4 + BA =

Anzeige: 15E. HEX

**Aufgabe:**  $8 \times 3$ **Tastenfolge:**  $8 \times 3 =$ **Anzeige:****18. HEX****Aufgabe:**  $(12 + D) \times B$ **Tastenfolge:**  $c\text{-ce} \quad 1 \quad 2 \quad + \quad D \quad ) \quad \times \quad B \quad =$ **Anzeige:****155. HEX****Aufgabe:**  $43A - 3CB =$ **Tastenfolge:**  $43A - 3CB =$ **(Total)****Tastenfolge:**  $c\text{-ce} \quad x\text{-m} \quad 4 \quad 3 \quad A \quad - \quad 3 \quad C \quad B \quad M+$ **Anzeige:****6F. HEX****Tastenfolge:**  $A \quad 3 \quad 8 \quad - \quad 2 \quad F \quad B \quad M+$ **Anzeige:****73D. HEX****Tastenfolge:**  $R\text{-M}$ **Anzeige:****7AC. HEX**

## 2.1.2 NORMALE BERECHNUNGEN

Die nachfolgenden Beispiele zeigen die Bedienung des Rechners bzw. die Tastenfolge für die Grundrechenfunktionen.

Bevor damit begonnen wird, die Beispiele nachzuvollziehen, sollte, um eine gemeinsame Ausgangsbasis sicherzustellen, eine kurze Überprüfung der Grundeinstellung des Computers erfolgen.

**POWER-Schalter auf ON schalten**

(Falls der Rechner automatisch abgeschaltet hat, zum Einschalten die **ON-BRK**-Taste drücken)

**CAL-Taste (grün) drücken**

**c-ce-Taste (rot) zweimal drücken**

Das Anzeigefeld sollte nun nur die Anfangsinformationen enthalten. Ist dies nicht der Fall, die Grundeinstellung des Computers gemäß 2.1.1.3 (Seite 53) durchführen.

### 2.1.2.1 Addition und Subtraktion

Aufgabe: Rechnen von 123 + 654

Tastenfolge

**r** **a** **b**  
1 2 3

**+**  
^

**σx** **Sx** **Σ**  
6 5 4

**=**

Anzeige

123.

123.

654.

777.

$$(123 + 654 = 777)$$

Aufgabe:  $12 + 45,6 - 32,1 + 789 - 741 + 213$

Tastenfolge: 12 **+** 45.6 **-** 32.1 **+** 789 **-** 741 **+** 213 **=**

Ergebnis: 286.5

### 2.1.2.2 Multiplikation und Division

Aufgabe:  $841 \times 586 / 0,12$

Tastenfolge: 841 **X** 586 **÷** .12 **=**

Ergebnis: 4106883.333

Aufgabe:  $427 + 54 \times 32 / 7 - 39 \times 2$

Tastenfolge: 427 **+** 54 **X** 32 **÷** 7 **-** 39 **X** 2 **=**

Ergebnis: 595.8571429

Anmerkung: Multiplikation und Division haben Vorrang vor Addition und Subtraktion. Die Vorrangordnung wird beachtet. Der Computer führt intern zuerst die Multiplikation und Division durch (siehe auch 2.1.2.7, Seite 66).

### Multiplikationen mit einer Konstanten

Der zuerst eingegebene Wert kann als Konstante verwendet werden.

Aufgabe:  $5 \times 3 = ?$ ,  $10 \times 3 = ?$ , usw.

Tastenfolge: 3 **X** 5 **=**

Ergebnis: 15

Tastenfolge: 10 **=**

Ergebnis: 30

**Divisionen mit einer Konstanten**

Der als zweites eingegebene Wert kann als Konstante verwendet werden.

Aufgabe:  $15 / 3 = ?$ ,  $30 / 3 = ?$ , usw.

Tastenfolge: 15  $\div$  3  $=$

Ergebnis: 5

Tastenfolge: 30  $=$

Ergebnis: 10

**Hinweis:** Bei Kettenrechnungen bleiben die entsprechend der Vorrangordnung letzte Recheninstruktion und der jeweilige numerische Wert erhalten und können damit für weitere Berechnungen bzw. als Konstanten verwendet werden.

**Ausführter****Rechenvorgang****Konstante**

$$\underline{a + b \times c} = +bc \quad (\text{Konstanten-Addition})$$

$$\underline{a \times b \div c} = \div c \quad (\text{Konstanten-Division})$$

$$\underline{a \div b \times c} = \frac{a}{b} \times \quad (\text{Konstanten-Multiplikation})$$

$$\underline{a \times b - c} = -c \quad (\text{Konstanten-Subtraktion})$$

**2.1.2.3 Potenz- und Wurzelfunktion**

Aufgabe:  $20^2$

Tastenfolge: 20  $x^2$

Ergebnis: 400

Aufgabe:  $3^3$

Tastenfolge: 3  $y^x$  3  $=$

Ergebnis: 27

Aufgabe:  $\sqrt{25}$

Tastenfolge: 25  $\square$

Ergebnis: 5

Aufgabe: 3. Wurzel von 27

Tastenfolge: 27 **SHIFT**  $\sqrt[3]{ }$

Ergebnis: 3

Aufgabe: 4. Wurzel von 81

Tastenfolge: 81 **SHIFT**  $\sqrt[4]{ }$  4  $=$

Ergebnis: 3

### 2.1.2.4 Prozentrechnung

Aufgabe: 45% von 2780 ( $2,780 \times \frac{45}{100}$ )

Tastenfolge: 2780 **[X]** 45 **[SHIFT]** **[A%]**

Ergebnis: 1251

### Prozentuale Zunahme/Abnahme

Der Ausgangswert (= 100%) muß als zweiter Wert, d.h. nach Drücken der **[ - ]**-Taste eingegeben werden.

Aufgabe: Anstieg von 473 auf 547, Prozentuale Zunahme?

Tastenfolge: 547 **[ - ]** 473 **[SHIFT]** **[A%]**

Ergebnis: 15.6448203

Aufgabe: Rückgang von 547 auf 473, Prozentuale Abnahme?

Tastenfolge: 473 **[ - ]** 547 **[SHIFT]** **[A%]**

Ergebnis: -13.52833638

### 2.1.2.5 Reziprok-Rechnung

Aufgabe:  $1/6 + 1/7$

Tastenfolge: 6 **[1/X]** **[+]** 7 **[1/X]** **[=]**

Ergebnis: 0.309523809

### 2.1.2.6 Speicherrechnung

Der zur Verfügung stehende Speicher wird über die blau beschrifteten Tasten **[RM]** **[M+]** angesprochen. Bevor mit dem Speicher gearbeitet wird, muß dessen Inhalt ggf. erst gelöscht werden. Der Speicherinhalt bleibt auch bei ausgeschaltetem Rechner erhalten. Die Belegung des Speichers zeigt das Symbol **[M]** in der rechten oberen Ecke des Displays an. Zum Löschen des Speichers die Tasten **[C-CE]** und **[Z-M]** drücken.

Das Addieren eines Rechenergebnisses oder eines eingegebenen Werts zum Speicherinhalt erfolgt durch Drücken der **[M+]**-Taste. Das Drücken der **[RM]**-Taste bringt den Speicherinhalt am Display zur Anzeige.

Tastenfolge	Anzeige
12 <b>[+]</b> 5 <b>[M+]</b>	17
<b>[C-CE]</b>	0
<b>[RM]</b>	17
8 <b>[÷]</b> 2 <b>[M+]</b>	4
<b>[RM]</b>	21 (Speicherinhalt)

Soll ein Rechenergebnis oder ein eingegebener Wert vom Speicherinhalt subtrahiert werden, muß vor dem Betätigen der **M+**-Taste durch Drücken der **±/-**-Taste das Vorzeichen gewechselt werden.

## Tastenfolge

2 **+** 5 **=** **±/-** **M+****C-CE****RM**

## Anzeige

-7

0

14

Das Drücken der **±/-**-Taste überträgt den im Display angezeigten Wert in den Speicher. Der Speicherinhalt wird überschrieben.

## Tastenfolge

12 **X** 2 **=** **±/-** **M+****C-CE****RM**

## Anzeige

24

0

24

## 2.1.2.7 Vorrangordnung und Verwendung der "Klammern"-Tasten

Die Verwendung der Tasten **(** und **)** ist dann zwingend notwendig, wenn eine Serie von Einzelberechnungen zu einem Rechenvorgang zusammengefaßt werden soll und dabei die Vorrangordnung der Algebra außer Kraft gesetzt werden muß.

Sobald die **(**-Taste gedrückt wird, erscheint in der rechten oberen Ecke des Anzeigefelds das **(**-Symbol, welches so lange sichtbar bleibt, bis alle geöffneten Klammern wieder geschlossen wurden. Das Schließen der Klammer(n) über die **)**-Taste kann unterbleiben, wenn unmittelbar folgend die Tasten **=** oder **M+** zu drücken sind, da diese die gleiche Funktion erfüllen.

Die Berechnungen in der Klammer haben Priorität vor allen anderen Berechnungen. Die Klammerfunktion kann in einer Ebene bis zu fünfzehnmal verwendet werden. Als erstes werden die Berechnungen der innersten Klammer ausgeführt.

Aufgabe:  $12 + 42 \div (8 - 6)$

Tastenfolge: 12 **+** 42 **÷** **(** 8 **-** 6 **)** **=**

Ergebnis: 33

Aufgabe:  $126 \div [(3 + 4) \times (3 - 1)]$

Tastenfolge: 126 **÷** **(** **(** 3 **+** 4 **)** **×** **(** 3 **-** 1 **)** **=**

**)** **)** **=**

Ergebnis: 9

Anmerkung: Das Schließen der beiden Klammern unmittelbar vor der **=**-Taste (oder **M+**-Taste) kann unterbleiben.

## Vorrangordnung

Da der Computer die Vorrangordnung der einzelnen Berechnungen beachtet, ist es möglich, mathematische Formeln direkt, d.h., ohne daß Klammern eingefügt oder Formelumstellungen vorgenommen werden müssen, einzugeben. Die Vorrangordnung bzw. die Reihenfolge, nach der die Berechnungen in einer längeren Rechenformel ausgeführt werden, ist wie folgt:

- (1) Funktionen wie sin oder  $x^2$   
(2)  $y^x$ ,  $\sqrt[x]{y}$   
(3)  $x$ ,  $\div$  (Berechnungen, die gleiche Priorität besitzen, werden  
der Reihe nach ausgeführt)  
(4)  $+$ ,  $-$   
(5)  $=$ , M+, Δ%

Beispiel: Tastenfolge und Reihenfolge der Berechnungen bei

$$5 + 2 \times \sin 30 + 24 \times 5^3 = 3006$$

### Winkeleinheit DEG

Tastenfolge: 5 **[+]** 2 **[X]** 30 **[sin]** **[+]** 24 **[X]** 5 **[y<sup>x</sup>]** 3 **[=]**

Die Nummern ① ~ ⑥ geben die Reihenfolge der Berechnungen an.

Um die Berechnungen gemäß der Vorrangordnung bzw. der Verwendung von Klammern ausführen zu können, ist der Computer mit zusätzlichen Speichern ausgestattet. Diese ermöglichen es, daß mathematische Formeln, die bis zu acht Berechnungsebenen erforderlich machen, direkt eingegeben und berechnet werden können.

Die Klammerfunktion, welche ebenfalls eine Berechnungsebene ergibt, kann, solange keine weiteren Ebenen notwendig sind, bis zu fünfzehnmal verwendet werden.

$$a \times (((b - c \times (((d + e) \times f) \div g) \dots$$



- Bis zu 15 Klammern sind möglich

Funktionen mit nur einer Variablen wie ( $x^2$ ,  $1/x$ ,  $n!$ ,  $\rightarrow$ DEG,  $\rightarrow$ DMS, usw.) werden sofort nach dem Drücken der Taste berechnet und benötigen somit keine Ebene bzw. keinen Speicher.

**Beispiel: Berechnungsebenen bzw. Zwischenspeicher (ohne Klammerfunktion)**

1  $+$  2 =

(3) 1 Ebene

1  $+$  2  $\times$  3 =

(7) 2 Ebenen

1  $+$  2  $\times$  3  $y^x$  4 =

(163) 3 Ebenen

1  $+$  2  $\times$  3  $y^x$  4  $\div$  5 =

(33,4) 3 und anschl. 2 Ebenen

②

Mit Drücken der  $y^x$ -Taste sind 3 Ebenen erreicht. Nach Drücken der  $\div$ -Taste werden die Berechnungen " $y^x$ " mit höherer Priorität als " $x$ " mit gleicher Priorität als " $/$ " ausgeführt. Es bleiben somit nach Drücken der  $+$ -Taste noch 2 Ebenen erhalten. Nach Drücken der =-Taste wird zuerst " $/$ " (höhere Priorität) und anschließend " $+$ " ausgeführt.

**Beispiel: Berechnungsebenen bzw. Zwischenspeicher (mit Klammerfunktion)**

1  $+$  2  $\times$  3  $y^x$  ( 4  $\div$  5 ) =

① ② ③ ④

(5,81644937), 4 Ebenen

Die Berechnung in der Klammer wird zuerst ausgeführt. Anschließend folgt " $y^x$ ", dann " $x$ " und zuletzt "+".

1  $+$  2  $\times$  ( 3 - 4  $\div$  5 ) =

① ② ③ ④

(5,4) 4 und anschl. 3 Ebenen

Nach Drücken der ( )-Taste wird zuerst " $/$ ", dann " $-$ " ausgeführt. Auf das Drücken der =-Taste folgt " $x$ " und zuletzt "+".

## 2.1.3 WISSENSCHAFTLICHE BERECHNUNGEN

Bei der Durchführung von Berechnungen, die Winkelwerte enthalten, muß auf die Winkeleinheit, die verwendet wird, geachtet werden. Das Vorwählen der Winkeleinheit erfolgt durch Drücken der Tasten **SHIFT** und **DRG**. Die gewählte Winkeleinheit wird im Display angezeigt.

### 2.1.3.1 Trigonometrische und inverse trigonometrische Funktionen (Arcusfunktionen)

Aufgabe:  $\sin 30^\circ + \cos 40^\circ$

Tastenfolge: Winkeleinheit auf DEG

30 **sin** + 40 **cos** =

Ergebnis: 1,266044443

Aufgabe:  $\cos 0,25\pi$

Tastenfolge: Winkeleinheit auf RAD

.25 **X** **SHIFT** **π** **=** **cos**

Ergebnis: 0,707106781

Aufgabe:  $\arcsin 0,5$

Tastenfolge: Winkeleinheit auf DEG

.5 **SHIFT** **sin<sup>-1</sup>**

Ergebnis: 30

Aufgabe:  $\arccos -1$

Tastenfolge: Winkeleinheit auf RAD

1 **+/-** **SHIFT** **cos<sup>-1</sup>**

Ergebnis: 3,141592654 (Wert  $\pi$ )

Das Ergebnis einer inversen trigonometrischen Funktion kann nur in folgendem Bereich liegen:

$$\theta = \sin^{-1} x, \quad \theta = \tan^{-1} x$$

$$\text{DEG: } -90 \leq \theta \leq 90 [^\circ]$$

$$\text{RAD: } -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} [\text{rad}]$$

$$\text{GRAD: } -100 \leq \theta \leq 100 [\text{g}]$$

$$\theta = \cos^{-1} x$$

$$\text{DEG: } 0 \leq \theta \leq 180 [^\circ]$$

$$\text{RAD: } 0 \leq \theta \leq \pi [\text{rad}]$$

$$\text{GRAD: } 0 \leq \theta \leq 200 [\text{g}]$$

### 2.1.3.2 Hyperbel- und inverse Hyperbelfunktionen (Areafunktion)

Die Anwahl der Hyperbelfunktion wird im Display durch das Symbol "HYP" bzw. bei inverser Hyperbelfunktion durch "SHIFT HYP" angezeigt.

Aufgabe:  $\sinh 4$

Tastenfolge: 4 [hyp] [sin]

Ergebnis: 27,2899172

Aufgabe:  $\text{arsinh } 9$

Tastenfolge: 9 [shift] [arcs] [sin]

Ergebnis: 2,893443986

### 2.1.3.3 Logarithmische Funktionen

Aufgabe:  $\ln 21$ ,  $\log 173$

Tastenfolge: Natürlicher Logarithmus 21 [ln]

Ergebnis: 3,044522438

Allgemeiner Logarithmus 173 [log]

Ergebnis: 2,238046103

### 2.1.3.4 Exponentialfunktionen

Aufgabe:  $e^{3,0445}$ ,  $10^{2,238}$

Tastenfolge: Exponentialfunktion (e) 3,0445 [shift] [ex]

Ergebnis: 20,99952881 (siehe ln 21)

Allgem. Exponentialfunktion 2,238 [shift] [10<sup>x</sup>]

Ergebnis: 172,9816359 (siehe log 173)

### 2.1.3.5 Fakultät

Aufgabe:  $69!$  ( $n! = 1 \times 2 \times 3 \times \dots \times n$ )

Tastenfolge: 69 [shift] [n!]

Ergebnis: 1,711224524E 98 ( $1,711224524 \times 10^{98}$ )

Anmerkung: Bei Berechnungen der Fakultät kann es sehr leicht zur Überschreitung der Rechenkapazität des Computers kommen, was die Fehlermeldung "E" zur Folge hat.

Aufgabe:  ${}_8P_3 = \frac{8!}{(8-3)!}$

Tastenfolge: 8 [shift] [n!] [÷] ( 8 [ - ] 3 ) [shift] [n!] [=]

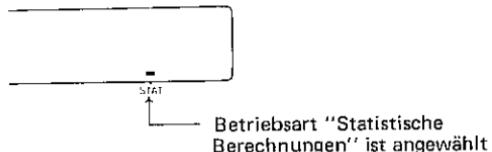
Ergebnis: 336

## 2.1.4 STATISTISCHE BERECHNUNGEN

Die Betriebsart "Statistische Berechnungen" wird durch Drücken der Tasten **SHIFT** (gelb) und **STAT** (unter der roten **C-CE**-Taste) angewählt. Falls sich der Rechner in Betriebsart "RUN" oder "PRO" befindet, muß zuerst in die Betriebsart "CAL" umgeschaltet werden.

Die Markierung (-) im rechten unteren Bereich des Anzeigefelds über der Beschriftung "STAT" gibt an, daß sich der Rechner in Betriebsart "Statistische Berechnungen" befindet.

Das Ausschalten der Betriebsart erfolgt durch Drücken der Tasten **SHIFT** und **STAT**.

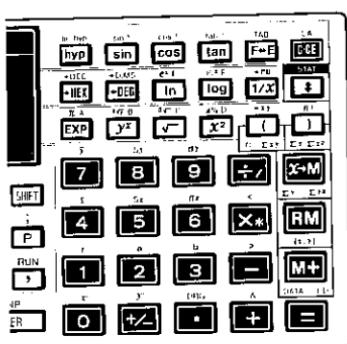


Wenn "STAT" angewählt ist, können die folgenden Berechnungen nicht durchgeführt werden:

- Speicherrechnungen
- Rechnungen, die Klammerausdrücke enthalten
- Koordinatenumwandlung
- Umwandlung "Hexadezimal-Dezimal"
- Rechnen mit Zahlen in hexadezimaler Schreibweise

In der Betriebsart "STAT" werden die in der folgenden Abbildung gekennzeichneten Tasten verwendet. Die Zweit-Funktion der verwendeten Tasten wird ebenfalls durch Drücken der **SHIFT**-Taste (gelb) angesprochen.

Tastenfeld für die Betriebsart "Statistische Berechnungen"



## Speicherung von Zwischen- und Endergebnissen

Bei der Durchfhrung von statistischen Berechnungen werden die nachfolgend aufgefhrten Ergebnisse automatisch den angegebenen, in der Betriebsart "BASIC" verwendeten Standardvariablen zugeordnet. Die Werte bleiben bei der Umschaltung in die Betriebsart "BASIC" erhalten und knnen somit direkt fr weitere Berechnungen verwendet oder in ein Programm bernommen werden.

Variable	Z	Y	X	W	V	U
Statistik	$n$	$\Sigma x$	$\Sigma x^2$	$\Sigma xy$	$\Sigma y$	$\Sigma y^2$

In diesem Zusammenhang ist jedoch folgendes zu beachten:

- Den Standardvariablen U bis Z wird ein Wert zugewiesen, d.h., ein gespeichertes Programm kann ggf. beeinträchtigt werden.
  - Um vorher eingegebene bzw. berechnete Statistikwerte zu löschen, muß, bevor mit einer neuen Berechnung begonnen wird, die Betriebsart "STAT" rückgesetzt und anschließend wieder angewählt werden. Die Korrektur von falsch eingegebenen Daten ist über die Tasten **SHIFT** **CD** möglich.

Falls mit den errechneten Statistikwerten weitere Berechnungen durchzuführen sind, sollte dies zweckmäßigerweise in der Betriebsart "RUN" erfolgen. Aus dem folgenden Beispiel ist die grundsätzliche Vorgehensweise ersichtlich. Das Rechnen in Betriebsart "RUN" ist in Abschnitt 2.2.1 ausführlich erläutert.

**Beispiel:** Berechnung der Summe der Quadrate ( $S^2$ ) der folgenden vier Werte 205, 221, 226 und 220.

Die Berechnungsformel für  $S^2$  lautet:

$$\begin{aligned} S^2 &= \Sigma(x - \bar{x})^2 \\ &= \Sigma x^2 - n\bar{x}^2 \\ &= \Sigma x^2 - \frac{1}{n} \end{aligned}$$

Die Werte für  $\Sigma x^2$  (X), n(Z) und  $\Sigma x$  (Y) werden in Betriebsart "STAT" ermittelt. Nach Umschaltung in "RUN" kann anschließend aus diesen Werten die Summe der Quadrate ( $S^2$ ) errechnet werden.

Tastenfolge:

Umschalten auf "STAT"

**CAL SHIFT STAT**

## Anzeige:

## Dateneingabe

205  221 226  220 

## Anzeige:

4.

Umschalten auf "RUN" und Berechnung von  $S^2$ 

## Anzeige:

&gt;

X - Y \* Y / Z -

246.

## Statistische Berechnungen mit einer Variablen

In den statistischen Berechnungen finden folgende Benennungen Verwendung:

 $n$  Anzahl der eingegebenen Daten $\Sigma x$  Gesamtsumme der eingegebenen Daten $\Sigma x^2$  Quadratsumme der eingegebenen Daten $\bar{x}$  Mittelwert der eingegebenen Daten  $\bar{x} = \frac{\Sigma x}{n}$  $S_x$  Standardabweichung mit Gesamtheitsparameter "n-1" (Stichproben-Standardabweichung)

$$S_x = \sqrt{\frac{\Sigma x^2 - n\bar{x}^2}{n-1}}$$

(Wird dann verwendet, wenn die eingegebenen Daten eine Auswahl bzw. Stichprobe aus der Gesamtheit darstellen.)

 $\sigma_x$  Standardabweichung mit Gesamtheitsparameter "n" (Standardabweichung der Gesamtheit)

$$\sigma_x = \sqrt{\frac{\Sigma x^2 - n\bar{x}^2}{n}}$$

(Wird dann verwendet, wenn die eingegebenen Daten als Gesamtheit aufzufassen sind bzw. die Stichproben die Gesamtheit ergeben.)

Bei statistischen Berechnungen mit einer Variablen werden die Daten wie folgt eingegeben:

Wert  oder, falls mehrmals der gleiche Wert einzugeben ist,Wert  Häufigkeit

Aufgabe: Berechnen der Standardabweichung  $S_x$ , des Mittelwerts  $\bar{x}$  und der Varianz ( $S_x^2$ ) der folgenden Daten:

Wert	35	45	55	65
Häufigkeit	1	1	5	2

Nach jeder Eingabe wird im Display die Anzahl der bisher eingegebenen Daten (n) angezeigt.

Tastenfolge:

35 [DATA]  
45 [DATA]  
55 x 5 [DATA]  
65 x 2 [DATA]

Anzeige:

1.  
2.  
7.  
9.

[SHIFT] [TAB] 2 (Einst. auf zwei Nachkommastellen)

Mittelwert	[SHIFT] [x̄]	53.89
Standardabweichung	[SHIFT] [Sx]	9.28
Varianz	[x̄²]	86.11

**Korrektur eingegebener Daten (CD):** Die zuletzt eingegebenen Werte 65 x 2 sind falsch und müssen auf 60 x 2 berichtigt werden.

Tastenfolge:

65 [x̄] 2 [SHIFT] [CD]  
60 [x̄] 2 [DATA]

Anzeige:

7.00  
9.00

Mittelwert	[SHIFT] [x̄]	52.78
Standardabweichung	[SHIFT] [Sx]	7.95
Varianz	[x̄²]	63.19

### Statistische Berechnungen mit zwei Variablen und Lineare Regression

Zusätzlich dazu, daß die statistischen Funktionen für zwei Variable (x und y) vorhanden sind und über die **[Exy]**-Taste die Summe der xy-Werte errechnet werden kann, ist es in den statistischen Berechnungen für zwei Variable möglich, eine Beziehung (Korrelation) zwischen den beiden Datenfolgen herzustellen.

Jedes eingegebene Datenpaar besteht aus einem x- und y-Wert. Aus den Datenpaaren läßt sich eine Regressionslinie ableiten. Da die Beziehung zwischen den Datenpaaren über eine gerade Linie hergestellt wird, spricht man von einer "Linearen Regression".

Die Lineare Regression weist drei wichtige Werte auf: r, a und b. Die Gleichung der geraden Linie ist  $y = a + bx$ , wobei a den Punkt beschreibt, an dem die Linie die y-Achse schneidet, während b die Steilheit angibt.

Der Korrelationskoeffizient r ermöglicht eine Aussage darüber, mit welcher Genauigkeit ein Bezug zwischen den beiden Datenfolgen hergestellt werden kann. Eine perfekte Korrelation zwischen zwei Werten ist bei  $r = 1$  bzw.  $r = -1$  (negative Korrelation) gegeben, d.h., daß in diesem Fall, wenn der Wert der einen Variablen bekannt ist, der andere Wert mit einer Genauigkeit von 100% ermittelt werden kann. Je weiter der Korrelationskoeffizient r von 1 entfernt ist, um so ungenauer sind die ermittelten Werte.

Die nachfolgende Tabelle soll eine Bewertung des Korrelationskoeffizienten r ermöglichen.

	Korrelationskoeffizient (r)	Genauigkeit
Positive Korrelation	+ 0,8 ... + 1,0	sehr hoch
	+ 0,6 ... + 0,8	hoch
	+ 0,4 ... + 0,6	mittelmäßig
	+ 0,2 ... + 0,4	gering
	- 0,2 ... + 0,2	-----
	- 0,2 ... - 0,4	gering
Negative Korrelation	- 0,4 ... - 0,6	mittelmäßig
	- 0,6 ... - 0,8	hoch
	- 0,8 ... - 1,0	sehr hoch

#### Berechnungsformeln:

$$r = \frac{S_{xy}}{\sqrt{S_{xx} \cdot S_{yy}}}$$

$$\left. \begin{array}{l} S_{xx} = \sum x^2 - \frac{(\sum x)^2}{n} \\ S_{yy} = \sum y^2 - \frac{(\sum y)^2}{n} \\ S_{xy} = \sum xy - \frac{\sum x \cdot \sum y}{n} \end{array} \right\}$$

$$\left. \begin{array}{l} a = \bar{y} - b \bar{x} \\ b = \frac{S_{xy}}{S_{xx}} \end{array} \right\} \quad y = a + bx$$

**Beispiel 1:** Auf einer Prüfung erzielten sechs Prüfungskandidaten folgende Ermittlung der voraussichtlichen Punktezahl in der Englischprüfung aus der in der Mathematikprüfung erreichten Punktezahl.

Die Grundlage der statistischen Berechnung bilden die erreichten Punktezahlen von sechs zufällig ausgewählten Prüfungskandidaten.

Prüfungskandidat Nr. n	Punkte in Math. x	Punkte in Englisch y
1	82	79
2	53	50
3	61	87
4	74	96
5	51	73
6	51	73

Tastenfolge:	Anzeige:
82 [x,y] 79 DATA	1.
53 [x,y] 50 DATA	2.
61 [x,y] 87 DATA	3.
74 [x,y] 96 DATA	4.
51 [x,y] 73 X 2 DATA	6.

Anmerkung: Mehrere identische Werte für x, y können über die **[X]**-Taste eingegeben werden.

Tastenfolge:	Anzeige:
SHIFT TAB 2	
SHIFT r	0.57
SHIFT a	34.26
SHIFT b	0.68

Der Korrelationskoeffizient  $r = 0,57$  besagt, daß eine mittelmäßige Genauigkeit zu erwarten ist.

Die Gleichung der Geraden lautet für dieses Beispiel  $y = 34,62 + 0,68x$ .

Tastenfolge:	Anzeige:
90 SHIFT y'	95.33

Aufgrund dieser statistischen Berechnung kann ein Prüfungskandidat, der in der Mathematikprüfung 90 Punkte erreichte, damit rechnen, in der Englischprüfung 95 Punkte zu erreichen.

**Beispiel 2:**

Kann aus dem Gewicht, das ein Mann im Alter von 65 Jahren besitzt, auf die Gesamtlebensdauer geschlossen werden? Im Jahre 1950 wurden zehn Männer von annähernd gleicher Körpergröße (1,80 m) für ein Experiment ausgewählt, das Aufschluß darüber geben sollte, ob das Gewicht einen Einfluß auf die Gesamtlebensdauer hat.

Nummer	1	2	3	4	5	6	7	8	9	10
Lebensdauer	72	67	69	85	91	68	77	74	70	82
Gewicht mit 65	83	102	91	77	77	88	79	79	90	78

Tastenfolge: "STAT", **SHIFT TAB** •  
 83 **[x,y]** 72 **[DATA]**  
 102 **[x,y]** 67 **[DATA]**

(alle Daten eingeben)

**SHIFT r** -0.787476214

Der Korrelationsfaktor **r** gibt eine relative hohe negative Korrelation an. Ein höheres Gewicht bedeutet somit eine kürzere Lebensdauer.

Zum Zeichnen der Regressionslinie sind die Werte für **a** und **b** erforderlich.

**SHIFT a** 140.977975 (y-Achse)  
**SHIFT b** -0.775805391 (Steilheit)

Wie alt wird ein ca. 1,80 m großer Mann, der 86 kg wiegt?

86 **SHIFT x'** 70.86567797 Jahre

Um 90 Jahre alt zu werden, sollte ein 1,80 m großer Mann wie schwer sein?

90 **SHIFT y'** 71.15548981 kg

Wie schwer müßte man sein, um 150 Jahre alt zu werden? Aus dieser letzten Frage wird deutlich, daß die Lineare Regression nur in einem bestimmten eingeschränkten Bereich brauchbare Ergebnisse liefert.

## 2.1.5 RECHENBEREICH

### Arithmetische Berechnungen:

Erster Operand  
 Zweiter Operand  
 Rechenergebniss }  $\pm 1 \times 10^{-99}$  bis  $\pm 9,999999999 \times 10^{99}$  und 0

## Wissenschaftliche Berechnungen:

Funktion	Rechenbereich	Anmerkung
$\sin x$	DEG: $ x  < 1 \times 10^{10}$ RAD: $ x  < \frac{\pi}{180} \times 10^{10}$ GRAD: $ x  < \frac{10}{9} \times 10^{10}$	
$\cos x$		
$\tan x$	Für $\tan x$ gelten folgende Einschränkungen DEG: $ x  = 90 (2n - 1)$ RAD: $ x  = \frac{\pi}{2} (2n - 1)$ n = Ganze Zahl GRAD: $ x  = 100 (2n - 1)$	
$\sin^{-1} x$	$-1 \leq x \leq 1$	
$\cos^{-1} x$		
$\tan^{-1} x$	$ x  < 1 \times 10^{100}$	
$\ln x$		
$\log x$	$1 \times 10^{-99} \leq x < 1 \times 10^{100}$	( $\ln x = \log_e x$ )
$e^x$	$-1 \times 10^{100} < x \leq 230.2585092$	$e \approx 2.718281828$
$10^x$	$-1 \times 10^{100} < x < 100$	
$y^x$	<ul style="list-style-type: none"> <li>• <math>y &gt; 0: -1 \times 10^{100} &lt; x \log y &lt; 100</math></li> <li>• <math>y = 0: x &gt; 0</math></li> <li>• <math>y &lt; 0: x</math>: ganzzahlig oder <math>1/x</math>: ungerade <math>-1 \times 10^{100} &lt; x \log  y  &lt; 100</math></li> </ul>	$y^x = 10^{x \cdot \log y}$
$\sqrt[x]{y}$	<ul style="list-style-type: none"> <li>• <math>y &gt; 0: -1 \times 10^{100} &lt; \frac{1}{x} \log y &lt; 100, x \neq 0</math></li> <li>• <math>y = 0: x &gt; 0</math></li> <li>• <math>y &lt; 0: x</math> oder <math>1/x</math>: ganzzahlig (<math>x \neq 0</math>) <math>-1 \times 10^{100} &lt; \frac{1}{x} \log  y  &lt; 100</math></li> </ul>	$\sqrt[x]{y} = 10^{\frac{1}{x} \cdot \log y}$
$\sqrt[3]{x}$	$ x  < 1 \times 10^{100}$	
$\sinh x$		
$\cosh x$		
$\tanh x$	$-227.9559242 \leq x \leq 230.2585092$	
$\sinh^{-1} x$	$ x  < 1 \times 10^{50}$	
$\cosh^{-1} x$	$1 \leq x < 1 \times 10^{50}$	
$\tanh^{-1} x$	$ x  < 1$	
$\sqrt{x}$	$0 \leq x < 1 \times 10^{100}$	
$x^2$	$ x  < 1 \times 10^{50}$	

Funktion	Rechenbereich	Anmerkung
$\frac{1}{x}$	$ x  < 1 \times 10^{100}$ $x \neq 0$	
$n!$	$0 \leq n \leq 69$ ( $n = \text{Ganze Zahl}$ )	
$\rightarrow \text{DEG}$	$ x  < 1 \times 10^{100}$	
$\rightarrow \text{D.MS}$	$ x  < 1 \times 10^{100}$	
$\text{HEX} \rightarrow \text{DEC}$	$0 \leq x \leq 2540\text{BE3FF}$ $\text{FDABF41C01} \leq x \leq \text{FFFFFFFFF}$	$x$ ist in "HEX" eine ganze Zahl
$\text{DEC} \rightarrow \text{HEX}$	$ x  \leq 9999999999$	$x = \text{ganzzahlig}$
$x, y \rightarrow r, \theta$	$(x^2 + y^2) < 1 \times 10^{100}$ $\frac{y}{x} < 1 \times 10^{100}$	$r = \sqrt{x^2 + y^2}$ $\theta = \tan^{-1} \frac{y}{x}$
$r, \theta \rightarrow x, y$	$r < 1 \times 10^{100}$ $ r \sin \theta  < 1 \times 10^{100}$ $ r \cos \theta  < 1 \times 10^{100}$	$x = r \cos \theta$ $y = r \sin \theta$ $\theta$ ist in derselben Bedingung wie $x$ von $\sin x, \cos x$
Data CD	$ x  < 1 \times 10^{50}$ $ y  < 1 \times 10^{50}$ $ \sum x  < 1 \times 10^{100}$ $\sum x^2 < 1 \times 10^{100}$ $ \sum y  < 1 \times 10^{100}$ $\sum y^2 < 1 \times 10^{100}$ $ \sum xy  < 1 \times 10^{100}$ $ n  < 1 \times 10^{100}$	
$\bar{x}$	$n \neq 0$	
Sx	$n \neq 1$ $0 \leq \frac{\sum x^2 - n\bar{x}^2}{n-1} < 1 \times 10^{100}$	
$\sigma x$	$n \neq 0$ $0 \leq \frac{\sum x^2 - n\bar{x}^2}{n} < 1 \times 10^{100}$	
$\bar{y}$	$n \neq 0$	
Sy	$n \neq 1$ $0 \leq \frac{\sum y^2 - n\bar{y}^2}{n-1} < 1 \times 10^{100}$	

Funktion	Rechenbereich	Anmerkung
Statistische Berechnungen	$\sigma y$ $n \neq 0$ $0 \leq \frac{\sum y^2 - n\bar{y}^2}{n} < 1 \times 10^{100}$	
	$r$ $n \neq 0$ $0 <  (\sum x^2 - n\bar{x}^2) \cdot (\sum y^2 - n\bar{y}^2)  < 1 \times 10^{100}$ $ \frac{\sum xy - \frac{\sum x \cdot \sum y}{n}}{n}  < 1 \times 10^{100}$ $ \frac{\sum xy - \frac{\sum x \cdot \sum y}{n}}{\sqrt{(\sum x^2 - n\bar{x}^2) \cdot (\sum y^2 - n\bar{y}^2)}}  < 1 \times 10^{100}$	
	$b$ $n \neq 0$ $0 <  \sum x^2 - n\bar{x}^2  < 1 \times 10^{100}$ $ \frac{\sum xy - \frac{\sum x \cdot \sum y}{n}}{n}  < 1 \times 10^{100}$ $ \frac{\sum xy - \frac{\sum x \cdot \sum y}{n}}{\sum x^2 - n\bar{x}^2}  < 1 \times 10^{100}$	
	$a$ $a$ ist in derselben Bedingung wie $b$ , und $ \bar{y} - b\bar{x}  < 1 \times 10^{100}$	
	$y'$ $ a + bx  < 1 \times 10^{100}$	
	$x'$ $ \frac{y - a}{b}  < 1 \times 10^{100}$	

Die Genauigkeit beträgt in der Regel für die Fließkomma-Schreibweise  $\pm 1$  in der 10. Stelle und für die wissenschaftliche Schreibweise  $\pm 1$  in der 9. Nachkommastelle der Mantisse.

Die Berechnungsgenauigkeit sinkt jedoch in der Nähe des Singulär-Punktes und des Umkehrpunktes der Funktion ab.

Ferner ist zu beachten, daß bei Kettenrechnungen eine Kumulierung des Fehlers erfolgt und damit mit jedem Rechengang eine Verschlechterung der Genauigkeit einhergeht. Der gleiche Effekt tritt rechnerintern bei der Durchführung von Funktionen wie  $y^x$  und  $\sqrt[x]{y}$  auf.

## 2.2 SHARP PC-1450 – EINSATZ ALS BASIC-RECHNER

### **2.2.1 RECHNEN OHNE PROGRAMMUNTERSTÜTZUNG (RUN-MODE)**

Der Computer kann auch in der Programmiersprache BASIC programmiert werden. Dabei kann er auf zwei verschiedene Arten verwendet werden. Zum einen können Sie Ihre Berechnungen innerhalb eines Programms abarbeiten, zum anderen können die Anweisungen und Funktionen direkt eingegeben und ausgeführt werden (RUN-Mode).

Dabei kann man den Computer wie einen einfachen Taschenrechner benutzen. Um das Ergebnis einer Rechnung zu erhalten muß man anstelle der '='-Taste die **[ENTER]**-Taste drücken.

Die allgemeine Form einer Rechnung im RUN-Mode ist:

**[C-CE]** numerischer Ausdruck **[ENTER]**

Als 'numerischer Ausdruck' wird jede mathematische Formel bezeichnet, die einen Zahlenwert als Ergebnis hat.

#### **2.2.1.1 Grundrechnungsarten**

a) Addition

**[5] [0] [+]** **[5] [0]** **[ENTER]**

**100.**

b) Subtraktion

**[1] [0] [0] [-]** **[5] [0]** **[ENTER]**

**50.**

c) Division

**[3] [0] [0] [/]** **[5]** **[ENTER]**

**60.**

d) Multiplikation

**[6] [0] [\*]** **[1] [0]** **[ENTER]**

**600.**

e) Klammerrechnungen, Exponentiation

**(** **[6] [7]** **[5]** **[+]** **[6]**  
**[7] [5] [0]** **[)** **[/]** **[4]**  
**[5] [0] [0]** **[0]** **[ENTER]**

**0.165**

**Hinweise:**

1. Erst nach Drücken der **ENTER**-Taste wird die Rechnung oder das Kommando ausgeführt und das Ergebnis angezeigt.
2. Das Ergebnis der Rechnung kann für weitere Berechnungen verwendet werden. Dazu die Zahl nicht mit der **C-CE**-Taste löschen, sondern das nächste Operationszeichen der neuen Formel eingeben (+ - \* / ^).

**Beispiel:**

<b>3</b> <b>0</b> <b>0</b> <b>*</b> <b>1</b> <b>5</b> <b>0</b> <b>ENTER</b>	<b>45000.</b>
<b>*</b> <b>0</b> <b>1</b> <b>5</b>	<b>45000.*.15_</b>
<b>ENTER</b>	<b>6750.</b>
<b>-</b> <b>4</b> <b>0</b> <b>0</b> <b>0</b>	<b>6750.-4000_</b>
<b>ENTER</b>	<b>2750.</b>
<b>-</b> <b>1</b> <b>2</b> <b>2</b> <b>5</b> <b>ENTER</b>	<b>1525.</b>
<b>-</b> <b>2</b> <b>2</b> <b>0</b> <b>0</b> <b>ENTER</b>	<b>-675.</b>

### 2.2.1.2 Rechnengenauigkeit

Der Computer verarbeitet Zahlen mit einer Genauigkeit bis zu 10 Stellen. Dies führt bei einigen Rechnungen zu Rundungsfehlern. Die Genauigkeitsbeschränkung tritt auch bei der Exponentiation oder beim Addieren von Zahlen sehr unterschiedlicher Größe auf.

#### Beispiele:

Eingabe

A = 199 ^ 3

ENTER

Ausgabe

7880598.999

Eingabe

A - 199 \* 199 \* 199

ENTER

Ausgabe

-0.001

Eingabe

1E12+1-1E12

ENTER

Ausgabe

Ø.

### 2.2.1.3 Wissenschaftliche Schreibweise

Man kann Zahlen in den Computer auch im wissenschaftlichen Format eingeben. Damit ist es möglich, Zahlen bis zu einer Größenordnung von  $\pm 9.999999999E \pm 99$  einzugeben und zu verarbeiten.

Wird dieser Wertebereich überschritten, erscheint die Fehlermeldung **ERROR 2**.

Ist der Betrag einer Zahl kleiner als  $1 E - 99$ , so wird die Zahl auf Null gesetzt.

Wird eine Zahl mit einem mehr als zweistelligen Exponenten eingegeben, so werden nur die beiden zuletzt eingegebenen Ziffern bewertet.

#### Hinweis:

1. Wie im englischen Sprachraum üblich, wird nicht das Komma, sondern der Punkt zur Trennung von ganzem und gebrochenem Anteil von Zahlen verwendet.
2. Für den Operator "geteilt durch" wird ein Schrägstrich (/) anstelle des Doppelpunkts gesetzt.
3. Für "multipliziert mit" muß ein Stern (\*) gesetzt werden.
4. Als Zeichen für die Exponentation wird ein Dach (^) verwendet. Ist die Basis negativ, muß sie in Klammern gesetzt werden. Als Exponenten sind bei negativen Grundzahlen nur ganze Zahlen zulässig. Ansonsten erscheint die Fehlermeldung **ERROR 2**.
5. Sind mehr als 10 Ziffern für die Darstellung eines Ergebnisses erforderlich, wird es im wissenschaftlichen Format angezeigt. Das Anzeigeformat kann durch die BASIC-Anweisung USING verändert werden.

### 2.2.1.4 Editier-/Korrekturmöglichkeit

Tippfehler können während der Eingabe oder bei der Überprüfung der Eingaben korrigiert werden.

Mit den Cursor-Tasten **[◀]** und **[▶]** können Sie den Cursor nach links oder rechts bewegen, ohne daß dabei die eingegebene Formel verändert wird. Zur Korrektur stellen Sie den Cursor über das zu korrigierende Zeichen und überschreiben dieses mit dem neuen Zeichen.

Zum Löschen eines Zeichens den Cursor über das zu löschen Zeichen stellen und die Tasten **[SHIFT]** und **[DEL]** drücken.

Zum Einfügen von Zeichen drücken Sie die Tasten **[SHIFT]** und **[INS]**. Es wird ein Platzhalter (**\_**) in die Formel eingefügt, der durch ein neues Zeichen überschrieben werden kann. Durch Drücken der **[ENTER]**-Taste werden alle überflüssigen Platzhalter aus der Formel gelöscht.

Ist das Ergebnis einer Berechnung schon ausgegeben oder erschien bei der Berechnung eine Fehlermeldung, so kann die Ausgangsformel mit den Tasten **[↑]** oder **[↓]** wieder zu Anzeige gebracht und wie oben beschrieben geändert werden.

#### Hinweis:

BASIC-Schlüsselwörter wie LET, SIN, COS, PRINT usw. werden nach Betätigung der **[ENTER]**-Taste intern abgekürzt. Obwohl das Schlüsselwort unverändert auf der Anzeige erscheint, wird es beim Editieren durch ein einziges neues Zeichen vollständig überschrieben. Will man sehr lange Programmzeilen eingeben, kommt man vorerst nur bis zum 80. Zeichen. Drücken Sie nun die **[ENTER]**-Taste und stellen Sie den Cursor an das Zeilenende. Nun können Sie noch zusätzlich einige Zeichen eingeben. Die Eingabe muß wieder mit **[ENTER]** abgeschlossen werden.

### 2.2.1.5 Mathematische Funktionen/Klammerregeln

Der Computer verfügt über alle gebräuchlichen mathematischen Funktionen. Diese können, im Gegensatz zu seinen Vorgängern, bei denen die Eingabe nur über die Buchstabentasten möglich war, über Funktionstasten eingegeben und im BASIC verarbeitet werden.

#### Beispiel:

 Ein Bild eines Taschenrechners zeigt die Tastatur mit den Tasten sin, 0 und ENTER. Auf dem Display steht der Wert 0.

Die Reihenfolge der Einzelschritte bei der Berechnung eines komplexen mathematischen Ausdrucks wird durch Klammern festgelegt.

Wie in der Mathematik gibt es dabei eine implizite Klammerung, d.h., daß bestimmte Operationen vor anderen den Vorrang haben. Der Computer verfügt über 15 Klammerebenen.

#### Beispiele:

$$3 * 5 + 7 * 4$$

ist gleichbedeutend mit

$$(3 * 5) + (7 * 4)$$

Die Rangfolge der mathematischen Operatoren ist:

1. Klammern
2. Abruf von PI, MEM, Variablen
3. Funktionsoperationen in Bezug auf das folgende Argument
4. Exponentiation
5. Vorzeichen +, -
6. Multiplikation, Division
7. Addition, Subtraktion
8. Vergleichsoperationen
9. Logische Operationen

### 2.2.1.6 Hexadezimal-(Sedezimal-)Zahlen

Natürliche Zahlen im Bereich zwischen 0 und 65535 können auch in sedezimaler Form eingegeben werden.

Der Sedezimalzahl wird dabei ein &-Zeichen vorangestellt. Zur hexadezimalen Darstellung einer Zahl dienen die Ziffern zwischen 0 bis 9 und die Buchstaben A bis F.

**Beispiel:**

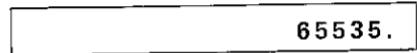
Ausgabe



Eingabe

Ausgabe



Die Ausgabe von Zahlen in Sedezimalform erfolgt mit der Tastenfolge:

 Dezimalzahl  

**Beispiel:**

123  

**Hinweise:**

1. In folgenden Fällen lässt sich eine Umwandlung nicht durchführen:
  - a) Wenn die Zahl eine Dezimalstelle hat, die bei Fließkommabetrieb nicht 0 ist.
  - b) Wenn eine Zahl in wissenschaftlicher Schreibweise dargestellt ist.
2. Die HEX-Funktion dient nur zur Ausgabe/Anzeige einer Dezimalzahl in sedezimaler Form. Das Ergebnis der Umwandlung kann nicht mehr weiterverarbeitet werden.

### 2.2.1.7 Textausdrücke

Textausdrücke sind Bestandteil der BASIC-Sprache. Sie können im RUN-Mode ohne Programmunterstützung eingegeben werden.

Man unterscheidet Textkonstanten, Textvariablen, Textfunktionen und zusammengesetzte Texte. Eine Textkonstante ist eine beliebige Zeichenfolge, die durch Anführungszeichen begrenzt ist, wobei die Anführungszeichen nicht Bestandteil der Textkonstante sind.

#### Beispiel:

Eingabe

**"SHARP"**

**ENTER**

Ausgabe

**SHARP**

Textvariablen können bis zu sieben Zeichen enthalten. Über die DIM-Anweisung lassen sich Textvariablen bis zu 80 Zeichen erzeugen.

#### Beispiel:

Eingabe

**A\$="SHARP"**

**ENTER**

Ausgabe

**SHARP**

Texte können mit Hilfe des "+"-Zeichens aneinandergefügt werden.

#### Beispiel:

Eingabe

**A\$="POCKET"**

**ENTER**

Ausgabe

**POCKET**

Eingabe

A\$+"\_COMPUTER"

ENTER

Ausgabe

POCKET COMPUTER

**Hinweis:**

Auch das Leerzeichen (Space) innerhalb eines Textes ist ein gültiges Zeichen.

**2.2.1.8 Logische Vergleichsausdrücke**

Der Computer kennt folgende Vergleichsausdrücke:

Mathematisches Symbol	BASIC	Deutsche Sprechweise
<	<	kleiner als
$\leq$	$\leq$	kleiner gleich
=	=	gleich
$\geq$	$\geq$	größer gleich
>	>	größer als
$\neq$	$\neq$	ungleich

Mit diesen Operatoren können sowohl zwei numerische als auch zwei Textausdrücke miteinander verglichen werden. Ist die Vergleichsaussage richtig, liefert der Rechner das Ergebnis "1", ist sie falsch, ist das Ergebnis "0". Im Zusammenhang mit der BASIC-Anweisung IF erhalten diese Vergleichsoperatoren eine besondere Bedeutung.

**Beispiele:**

Eingabe

C-CE

2 &lt; 1

ENTER

Ausgabe

Ø.

## b) Eingabe

**CCE** **1 <= 2**

**ENTER**

## Ausgabe

1.

## c) Eingabe

**CCE** **SIN 1 <.7**

**ENTER**

## Ausgabe

0.

(im RAD-Mode)

1.

(im DEG- und GRAD-Mode)

## d) Eingabe

**CCE** **"WERNER" > "ANTON"**

**ENTER**

## Ausgabe

1.

Die Zeichenketten "WERNER" und "ANTON" werden hier als logische Ausdrücke überprüft. Da die Zeichenketten aus den entsprechenden ASCII-Code-Ziffern bestehen, kann der Computer sie leicht vergleichen. Die Textausdrücke "WERNER" und "ANTON" werden entsprechend dem ASCII-Code auf die lexikographische Reihenfolge hin überprüft.

## Hinweis:

Da das Ergebnis eines logischen Ausdrucks eine Zahl ist, kann dieses als numerische Variable gespeichert und weiterverarbeitet werden.

## 2.2.2 SPRACHELEMENTE

### 2.2.2.1 Numerische Konstante

Eine numerische Konstante kann sein:

- eine ganze Zahl (positiv oder negativ)
- eine Dezimalzahl
- eine Zahl in wissenschaftlicher Schreibweise
- eine Sedenzimal-(Hexadezimal-)Zahl

Beispiele:

513

-2376

11.745

1.23456788E-12

&AA2B

### 2.2.2.2 Textkonstante

Eine Textkonstante ist eine beliebige Zeichenfolge, die durch Anführungszeichen ("") begrenzt wird.

Beispiele:

"SHARP"

"" (Textkonstante enthält Leerzeichen)

"" (Textkonstante der Länge 0)

### 2.2.2.3 Numerische Variable

Unter einer numerischen Variable versteht man den Speicherplatz, der zur Aufnahme des jeweiligen, zugehörigen Zahlenwertes bereitsteht.

Der Variablenname setzt sich aus einem Buchstaben oder einem Buchstaben mit einem in Klammern folgenden Indexwert zusammen. Dabei kann der Index ebenfalls durch eine Variable dargestellt werden.

Beispiele:

A

A(27)

B(1)

A(Z)

AB

A1

XY(2,4)

#### 2.2.2.4 Textvariable

Unter einer **Textvariable** versteht man den Speicherplatz, der zur Aufnahme der jeweiligen, zugehörigen Zeichenfolge bereitsteht.

Der Name der Textvariable ist im Prinzip gleich aufgebaut wie der der numerischen Variable mit einem zusätzlichen "\$"-Zeichen hinter dem Buchstaben.

### Beispiele:

A\$

B\$(2)

C\$ (Z)

### 2.2.2.5 Numerische Funktionen, Textfunktionen

Eine numerische Funktion setzt sich aus einem Operator und einem oder mehreren Parametern zusammen. Die Parameter werden hinter den Funktionsnamen gestellt und können je nach Funktion numerische oder Textausdrücke sein. Sind mehrere Parameter erforderlich, so werden diese in Klammern gesetzt und durch Komma trennt.

**Das Ergebnis einer numerischen Funktion ist ein Zahlenwert, das einer Textfunktion ein Zeichen bzw. eine Zeichenfolge.**

### Beispiele:

**LN60** numerische Funktion; natürlicher Logarithmus des numerischen Parameters 60

**ASC**"A" numerische Funktion; Ergebnis ist die numerische Konstante 65, die den ASCII-Code der Textkonstante "A" darstellt.

**CHR\$65** Textfunktion; Ergebnis ist die Textkonstante "A".

**MID\$("SHARP",2,2)** Textfunktion; Ergebnis ist die Textkonstante "HA".

#### **Hinweis:**

Funktionsoperatoren haben eine höhere Priorität als andere Operatoren.

#### Beispiel -

$$\sin A + B = (\sin A) + B$$

Soll der Sinus der Summe ( $A + B$ ) gebildet werden, so müssen Klammern gesetzt werden:

$$\sin(A + B)$$

### 2.2.2.6 Numerischer Ausdruck

Ein numerischer Ausdruck setzt sich aus einer numerischen Konstanten, Variablen, numerischen Funktion oder deren Verknüpfung durch die arithmetischen Operatoren +, -, \*, /, ^, AND, OR und NOT und die Zusammenfassung durch Klammern zusammen.

**Beispiele:**

15  
(-8)  
SIN 45  
A + B/C  
(C \* (A \* X+B) \* 5/(D \* C))+10

### 2.2.2.7 Textausdruck

Ein Textausdruck besteht aus einer Textkonstanten, Textvariablen oder einer Textfunktion und deren Verknüpfung durch das "+" -Zeichen. Das Ergebnis eines Textausdrucks ist eine Zeichenfolge.

**Beispiele:**

"A" + STR\$ A  
A\$ + "PC"

### 2.2.2.8 Logische Vergleichsausdrücke

Ein logischer Vergleichsausdruck ist der Vergleich zweier Ausdrücke (numerisch oder Text) durch die Operatoren:  $>$ ,  $<$ ,  $=$ ,  $\geq$ ,  $\leq$ ,  $\neq$

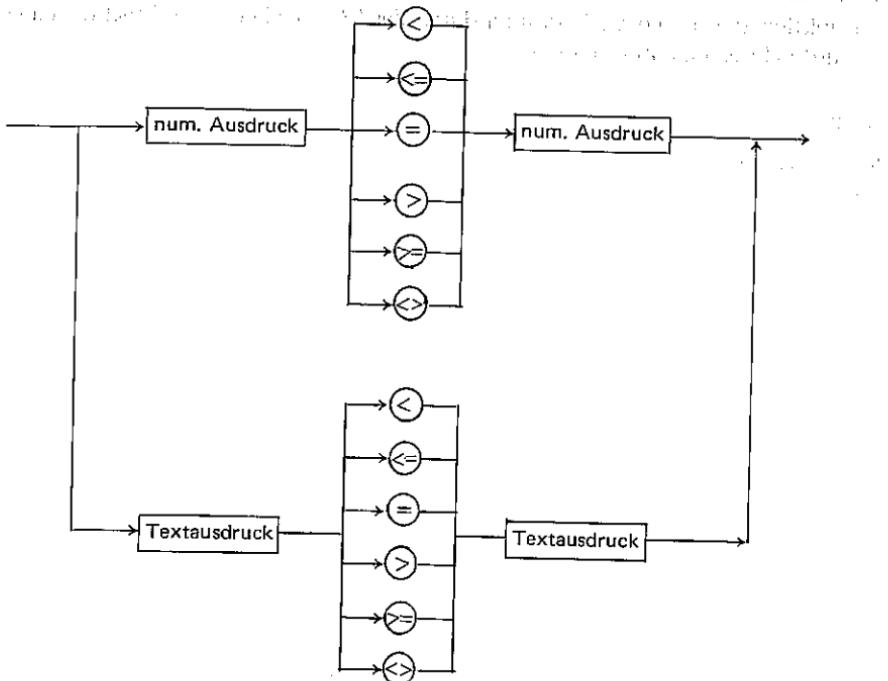
und die Verknüpfung solcher Vergleiche mit den Booleschen Operatoren AND, OR und NOT.

Operation	Mathematisches Symbol	BASIC
Negation	-	NOT
UND-Funktion	$\wedge$	AND
ODER-Funktion	$\vee$	OR

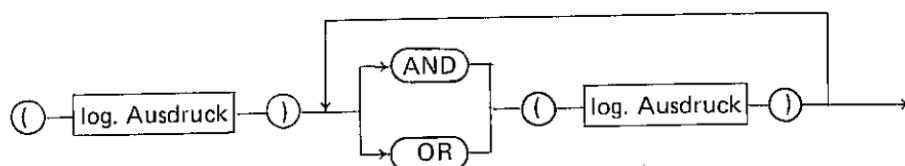
#### Syntax:

Logische Vergleichsausdrücke bestehen aus zwei Ausdrücken, die durch einen Vergleichsoperator verbunden sind. Der Vergleich kann mit numerischen oder textuellen Ausdrücken vorgenommen werden.

Die folgenden Diagramme zeigen die Struktur von Logischen Vergleichsausdrücken für numerische und textuelle Ausdrücke.



Logische Vergleichsausdrücke lassen sich wie folgt verknüpfen:



Ist die Vergleichsaussage richtig, so ist das Ergebnis des logischen Ausdrucks "1", ist sie falsch, ist das Ergebnis "0".

Werden Textausdrücke miteinander verglichen, so werden die Textausdrücke auf lexikographische Reihenfolge geprüft; diese richtet sich nach dem ASCII-Code (siehe Anhang).

Werden zwei Textausdrücke mit unterschiedlicher Länge verglichen, so werden beim Vergleich die fehlenden Stellen des kürzeren Ausdrucks mit dem ASCII-Zeichen Null aufgefüllt.

Logische Ausdrücke werden in der BASIC-Anweisung IF verwendet.

#### Beispiele:

	Ergebnis
$1 < 2$	1
$1 + 2 + 3 < 2 + 3 + 4$	1
$1 \geq 2$	0
$1 < 2 < 3$	1
"ANTON" < "WERNER"	1
"ANTON" > "ANTON 1"	0 ("1" ist größer als ASCII-Null)
$(1 < 2) \text{ AND } (3 < 4)$	1
$("ANTON" < "WERNER") \text{ OR }$	1
$("ANTON" = "ANTON 1")$	1

### 2.2.3 VARIABLEN

Variablen sind Bezeichner für Größen, deren Wert erst im Laufe des Programmablaufs festgelegt werden. Sie werden beim Rechner durch Speicherplätze realisiert, denen man unterschiedliche Daten durch Einlesen oder durch Auswerten eines bestimmten Ausdrucks zuordnen kann. Je nachdem, ob es sich bei der Variablen um einen numerischen oder um einen Textausdruck handelt, nennt man solche Variablen numerische Variablen oder Textvariablen.

Zur Unterscheidung zwischen numerischen und Textvariablen muß jeder Name einer Textvariablen mit dem "\$"-Zeichen enden. Beide Typen können als einfache oder indizierte Variable geschrieben werden. Zur leichteren Identifizierung erhalten sie einen Variablenamen als symbolische Adresse für den Speicherplatz. Mit Variablen lassen sich Gesetzmäßigkeiten unabhängig von ihrem jeweiligen Wert allgemein ausdrücken.

Im Computer stehen folgende zwei verschiedene Speicherbereiche für Variablen zur Verfügung.

- a) StandardvariablenSpeicher
- b) FeldvariablenSpeicher

Der StandardvariablenSpeicher ist ein begrenzter Speicherbereich und dient ausschließlich zur Aufnahme der 26 Standardvariablen.

Der FeldvariablenSpeicher ist Bestandteil des Hauptspeichers, der auch zur Aufnahme von Programmen dient. Die Kapazität des Hauptspeichers kann wahlweise für Programme und für Feldvariablen benutzt werden.

Folgende Variablentypen stehen zur Verfügung:

1. Standardvariablen
  - a) einfache numerische Variable
  - b) einfache Textvariable
  - c) indizierte numerische Variable
  - d) indizierte Textvariable
2. Feldvariablen
  - a) indizierte eindimensionale numerische Feldvariable (Vektor)
  - b) indizierte eindimensionale Textfeldvariable (Vektor)
  - c) indizierte zweidimensionale numerische Feldvariable (Matrizen)
  - d) indizierte zweidimensionale Textfeldvariable (Matrizen)

### 2.2.3.1 Standardvariable

Für die Standardvariablen stehen 26 reservierte Speicherbereiche zur Verfügung, die wahlweise als numerische oder als Textvariable belegt werden können.

Den Standardvariablen können also wahlweise numerische oder Textwerte zugeordnet werden. Bei der Wertabfrage der Standardvariablen muß der Variablenname dem Inhalt dieser Variablen entsprechen. Wird z.B. eine numerische Variable als Textvariable aufgerufen, und umgekehrt, so wird am Display die Fehlermeldung ERROR 9 angezeigt.

Standardvariable:

A = A\$ = A(1) = A\$(1)  
B = B\$ = A(2) = A\$(2)  
C = C\$ = A(3) = A\$(3)  
D = D\$ = A(4) = A\$(4)  
E = E\$ = A(5) = A\$(5)  
F = F\$ = A(6) = A\$(6)  
G = G\$ = A(7) = A\$(7)  
H = H\$ = A(8) = A\$(8)  
I = I\$ = A(9) = A\$(9)  
J = J\$ = A(10) = A\$(10)  
K = K\$ = A(11) = A\$(11)  
L = L\$ = A(12) = A\$(12)  
M = M\$ = A(13) = A\$(13)  
N = N\$ = A(14) = A\$(14)  
O = O\$ = A(15) = A\$(15)  
P = P\$ = A(16) = A\$(16)  
Q = Q\$ = A(17) = A\$(17)  
R = R\$ = A(18) = A\$(18)  
S = S\$ = A(19) = A\$(19)  
T = T\$ = A(20) = A\$(20)  
U = U\$ = A(21) = A\$(21)  
V = V\$ = A(22) = A\$(22)  
W = W\$ = A(23) = A\$(23)  
X = X\$ = A(24) = A\$(24)  
Y = Y\$ = A(25) = A\$(25)  
Z = Z\$ = A(26) = A\$(26)

### 2.2.3.2 Einfache Variablen

Einfache Variablen werden mit den Namen A bis Z als numerische Variable bzw. A\$ bis Z\$ als Textvariable aufgerufen.

Einer einfachen numerischen Variablen kann eine Zahl mit maximal 10 Stellen, einen zweistelligen Exponenten und die Vorzeichen zuordnen.

**Beispiele:** A = 123; A = SIN X; A = B + C; A = B \* C

A = 123

A = SIN X

A = B + C

A = B \* C

Einer Textvariablen kann man Zeichenfolgen bis zu sieben Zeichen, bestehend aus Buchstaben, Zahlen, Sonderzeichen und Leerstellen, zuweisen. Bei der Wertzuweisung muß der Text durch Anführungszeichen begrenzt werden.

**Beispiele:**

B\$="TEXT"

B\$="T E X T"

### 2.2.3.3 Indizierte Variablen

Indizierte Variablen werden mit den Namen A(1) bis A(26) als numerische bzw. mit A\$(1) bis A\$(26) als Textvariable aufgerufen.

Die indizierten Standardvariablen sind äquivalent zu den oben beschriebenen einfachen Standardvariablen. So entspricht z.B. die einfache Variable A der indizierten Variablen A(1), die einfache Variable B der indizierten Variablen A(2); usw.

Die genaue Zuordnung ist aus der Tabelle auf Seite 97 ersichtlich.

Die indirekte Adressierung (Indizierung) einer Variablen gestattet es, den Namen und die Adresse einer Variablen in Abhängigkeit vom Wert einer anderen Variablen, der z.B. ein Rechenergebnis sein kann, festzulegen. Außerdem kann der Index einer Variablen auch das Ergebnis eines numerischen Ausdrucks sein.

**Beispiele:**

LET A(A) = 1243

Hiermit wird eine numerische Variable definiert, deren Index gleich dem ganzzähligen Anteil der im Speicher A stehenden Zahl ist. Angenommen, der Wert der Variablen A ist 7, so wird der Variablen A(7), also der Variablen G, der Wert 1234 zugewiesen.

LET A(B/5) = 789

Hierbei wird eine numerische Variable definiert, deren Index gleich dem ganzzahligen Anteil des Quotienten aus B/5 ist.

Angenommen, der Wert der Variablen B ist 134, so ist der Quotient aus 134/5 = 26,8, und somit wird der Variablen A(26) = Z ein Wert von 789 zugeordnet.

LET A\$(A) = "TEXT"

Der Wert der numerischen Variablen A definiert den Index der Textvariablen A\$. Dabei muß der Wert der Variablen größer als 1 sein, da die Variable A\$(0) nicht existiert und die Variable A\$(1) der Variablen A\$ entspricht und die beiden, wie bereits erklärt, nicht gleichzeitig definiert werden können.

#### 2.2.3.4 Besonderheiten der Variablen A

Die Variable A kann als:

- a) Standardvariable A bzw. A\$
- b) Indizierte Standardvariable A(n) bzw. A\$(n)  
(n = 1 bis 26)
- c) Eindimensionale Feldvariable A(n) bzw. A\$(n)  
(n = 0 bis 255)
- d) Zweidimensionale Feldvariable A(n, n) bzw. A\$(n, n)  
(n = 0 bis 255)

definiert werden.

Dabei gelten folgende Grundsätze:

##### 1) Die Variable A ohne Dimensionierung

1. Diesen A-Variablen können nur numerische Werte mit zehnstelliger Mantisse, zweistelligem Exponenten und einem Vorzeichen bzw. Textwerte mit max. 7 Zeichen zugewiesen werden. Sie können nur alternativ als numerische oder Textvariablen definiert werden.
2. Der Variablenname A(0) ist unzulässig.
3. Für die indizierte Variable A(n), (n = 1 bis 26) = Standardvariablen A bis Z, wird kein Speicherplatz im Hauptspeicher reserviert.
4. Für die indizierte Variable A(n), (n = 27 bis 255) = eindimensionaler Feldvariable, wird ein Speicherplatz im Hauptspeicher reserviert. Der Platz wird automatisch durch den höchsten definierten Indexwert reserviert. Wird z.B. A(n), n = 50, aufgerufen, ist der A-Vektor mit den Elementen A(27) bis A(50) automatisch vereinbart.

5. Der A-Vektor wird bezüglich seiner Elementanzahl automatisch auf den Wert begrenzt, den er vor der Dimensionierung eines weiteren Vektors bzw. einer Matrix hätte.
6. Bei der ersten Verwendung von Indices größer als A(27) oder A\$(27) werden 7 Bytes für den Variablenamen und 8 Bytes für jede Variable verwendet.

**Beispiel:**

```
5: A(27) = 27
10: DIM B (10)
15: DIM A (40)    oder 15: A(40) = 40
```

Nach dem Starten des Programms wird die Fehlermeldung ERROR 3 angezeigt. Der A-Vektor wurde durch die Dimensionierung des B-Vektors auf ein Element, nämlich A(27), begrenzt.

Durch den Anruf von A(40) in Zeile 15 müßte ein zweiter A-Vektor definiert werden, was aber unzulässig ist.

7. Die Variablen A(n) bzw. A\$(n) ( $n = 1$  bis 26), können nicht gelöscht werden. Mit den Kommandos NEW bzw. CLEAR wird ihr Wert auf 0 (Null), bzw. auf das ASCII-Zeichen Null gesetzt. Variablen A(n) bzw. A\$(n) werden durch das Kommando RUN gelöscht.

**2) Die Variable A mit Dimensionierung**

1. Die dimensionierte Variable A wird behandelt wie in den Kapiteln 2.2.3.5. – 2.2.3.7. beschrieben ist.
2. Nach einer Dimensionierung der Variablen A kann auf die Standardvariablen nur noch direkt zugegriffen werden. Nach z.B. DIM A(26) entspricht A(1) nicht mehr der Standardvariablen A und A(26) nicht mehr der Standardvariablen Z.  
Das Element A(1) und die Standardvariable A haben nun verschiedene Speicherplätze.
3. Wird nach dem Aufruf einer undimensionierten Standardvariablen, z.B. A(2) = B eine Dimensionierung, z.B. DIM A(30) durchgeführt, wird ERROR 3 angezeigt.

### 2.2.3.5 Feldvariablen

Feldvariablen können als eindimensionale Felder (Vektoren) oder als zweidimensionale Felder (Matrizen) definiert werden, die wiederum unterschiedlich viele Elemente haben können.

Die Feldvariablen werden im Hauptspeicher abgelegt und müssen vor ihrem Aufruf wegen ihrer flexiblen Länge dimensioniert werden. Weitere Einzelheiten siehe unter dem BASIC-Befehl DIM Abschnitt 2.3).

Der Speicherplatzbedarf für eine Feldvariable setzt sich zusammen aus:

- 7 BYTES für den Variablenamen
- 8 BYTES für ein numerisches Element
- 16 BYTES für ein Textelement im Standardformat

### 2.2.3.6 Numerische Feldvariablen

Numerische Feldvariablen können sowohl als Vektoren als auch als Matrizen definiert werden. Jedem Element der Vektoren bzw. der Matrizen kann eine vollständige Zahl, bestehend aus einer zehnstelligen Mantisse, einem zweistelligen Exponenten und einem Vorzeichen zugeordnet werden.

Vektoren werden mit dem Vektorennamen und dem Index für ein Element aufgerufen.

Der Vektorennname kann aus einem oder aus zwei Zeichen bestehen. Das erste Zeichen muß ein Buchstabe sein, das zweite Zeichen kann ein Buchstabe oder eine Ziffer sein.

Buchstabenkombinationen, die ein BASIC-Schlüsselwort oder eine Funktion ergeben, sind als Variablenname nicht erlaubt, z.B. LN, ON, OR, PI etc.

**Beispiel:**

B(0)  
C1(3)  
DF(80)

Matrizen werden mit dem Matrizennamen und den beiden Indizes für ein Element aufgerufen.

Der Matrizenname kann aus einem oder aus zwei Zeichen bestehen. Das erste Zeichen muß ein Buchstabe sein, das zweite Zeichen kann ein Buchstabe oder eine Ziffer sein.

Buchstabenkombinationen, die ein BASIC-Schlüsselwort oder eine Funktion ergeben, sind als Variablenname nicht erlaubt, z.B. LN, ON, OR, PI etc.

**Beispiel:**

D(2,4) definiert einen Vektor mit 4 Elementen mit den Werten 2, 1, 3 und 4.  
**E1(1,1)** definiert eine 1x1-Matrix mit dem Wert 1. Es ist die einzige 1x1-Matrix  
**YZ(0,0)** definiert eine 2x2-Matrix mit den Werten 0,0, 0,0.

Insgesamt stehen für Vektoren und Matrizen folgende Namen zur Verfügung:

Alle Einzelbuchstaben oder alle Zweierkombinationen aus den Buchstaben A – Z und den Ziffern 0 – 9, wobei das erste Zeichen ein Buchstabe sein muß.

Die Indizes für die Elemente liegen im Intervall 0 – 255.

Buchstabenkombinationen, die ein BASIC-Schlüsselwort oder eine Funktion ergeben, sind als Variablenname nicht erlaubt, z.B. LN, ON, OR, PI etc.

Der Index für das Vektoren- und Matrizelement kann als numerische Konstante oder durch eine Variable definiert werden.

**Beispiele:** B(1,2) bzw. B(1,2,3) definiert den zweiten Wert des zweiten Feldnamens der Matrix B(1,2,3).  
**B(6)** bzw. **B(A)** definiert ebenfalls den sechsten Wert des Feldnamens B(A).  
**H1(3,5)** bzw. **H1(E,B)** definiert ebenfalls den Wert des Feldnamens H1(B).

Der Index wird durch den jeweiligen Wert der Variablen ausgedrückt.

Vektoren und Matrizen dürfen nicht mit dem gleichen Feldnamen und nicht zweimal definiert werden... Die Namen B(n) und B(n,n) sind also nicht gleichzeitig zulässig.

#### Hinweis:

Bei Verwendung der Feldvariablen 'A' beachten Sie das Kapitel 2.2.3.4.

### 2.2.3.7 Textfeldvariablen

Textfeldvariablen können wie die numerischen Feldvariablen als Vektor oder als Matrix definiert werden. Jedem Element der Vektoren oder Matrizen können im Standardformat Texte mit maximal 16 Zeichen zugeordnet werden. Durch eine entsprechende Dimensionierung (siehe DIM-Anweisung Seite 138) kann die Größe der Elemente zwischen 1 und 80 Zeichen frei festgelegt werden.

**Beispiel:** **DIM B\$(10) \* 4**  
In diesem Beispiel können für die Elemente des B\$-Vektors (B\$(0) bis B\$(10)) jeweils nur 4 Zeichen eingegeben werden.

Achtung: Ein Zeichen ist 1 Byte groß, das bedeutet, dass ein Zeichen 8 Bit benötigt. Eine Zeichenkette von 4 Zeichen benötigt daher 32 Bit.

Insgesamt stehen für die Vektoren und Matrizen folgende Feldnamen zur Verfügung:

Alle Einzelbuchstaben oder alle Zweierkombinationen aus den Buchstaben A – Z und aus den Ziffern 0 – 9, wobei das erste Zeichen ein Buchstabe sein muß.

Buchstabenkombinationen, die ein BASIC-Schlüsselwort oder eine Funktion ergeben, sind als Variablenname nicht erlaubt, z.B. LN, ON, OR, PI etc.

Die Indizes für die Elemente liegen im Intervall 0 – 255.

Vektoren werden mit dem Vektorennamen und dem Index für ein Element aufgerufen.

**Beispiele:**

B\$(0)

A5\$(0) \* 20

Z\$(80) \* 1

Matrizen werden mit dem Matrizenamen und den beiden Indizes für ein Element aufgerufen.

**Beispiele:**

D\$(2,4)

Z1\$(5,10) \* 12

Die Indizes für das Vektoren- und Matrizelement können als numerische Konstante oder als numerische Variable definiert sein, wobei der Index durch den Wert der jeweiligen Variablen bezeichnet wird.

**Beispiele:**

C\$(A)

AE\$(E,F)

Vektoren und Matrizen dürfen nicht mit dem gleichen Feldnamen und nicht zweimal definiert werden. Die Namen B\$(n) und B\$(n,n) sind also nicht gleichzeitig zulässig.

**Hinweis:**

Bei Verwendung der Texfeldvariablen 'A' beachten Sie das Kapitel 2.2.3.4.

## 2.2.4 ARITHMETISCHE FUNKTIONEN

Der Computer bietet eine große Anzahl von arithmetischen Standard-Funktionen. Hierzu gehören:

**Trigonometrische und ihre Umkehrfunktionen**

**Hyperbelfunktionen**

**Natürlicher und dekadischer Logarithmus und Umkehrfunktionen**

**Polarkoordinatenberechnungen**

Die Ergebnisse der Funktionen sind numerische Werte. Als Parameter ist im allgemeinen ein numerischer Ausdruck zugelassen.

Fast alle Funktionsnamen können über die Funktionstasten des Rechnerteils eingegeben werden:

Funktionsname	Funktion
ABS	Absolutwert
ACS	Arcuscosinus
AHC	Areacosinus
AHS	Area sinus
AHT	Areatangens
ASN	Arcussinus
ATN	Arcustangens
COS	Cosinus
CUR	Kubikwurzel
DEG	Sexagesimalumrechnung
DEGREE	Winkeleinheit Grad
DMS	Sexagesimalumrechnung
EXP	Exponentialumrechnung
FAC	Fakultät
GRAD	Winkeleinheit Neugrad
HCN	Hyperbelcosinus
HEX/DEC	Sedezimalumrechnung
HSN	Hyperbelsinus
HTN	Hyperbeltangens
INT	Ganzzahlfunktion
LN	Natürlicher Logarithmus
LOG	Dekadischer Logarithmus
MEM	Freier Speicherplatz
PI	Kreiskonstante PI
POL	Polarkoordinatenumrechnung
RADIAN	Winkeleinheit
RANDOM	Zufallsgeneratoranfangswert

RCP	<b>1/x</b>	Reziprokwert
REC	<b>SHIFT</b> <b>xy</b>	Polarkoordinatenumrechnung
RND		Zufallszahl
ROT	<b>SHIFT</b> <b>x<sup>y</sup></b>	Wurzelberechnungen (x-te Wurzel)
SGN		Vorzeichen
SIN	<b>sin</b>	Sinus
SQU	<b>x<sup>2</sup></b>	Quadratzahl
SQR	<b>√</b>	Quadratwurzel
TAN	<b>tan</b>	Tangens
TEN	<b>SHIFT</b> <b>10<sup>x</sup></b>	Exponentialfunktion $10^x$
^	<b>SHIFT</b> <b>^</b>	Potenzfunktion

## 2.2.5 TEXTFUNKTIONEN

Mit dem Computer können Sie nicht nur rein numerische Aufgaben lösen, sondern auch Texte verarbeiten. Die im folgenden Kapitel beschriebenen Funktionen sollen Ihnen diese Arbeit erleichtern. So können Sie beispielsweise Zeichenfolgen aus Texten heraustrennen und zu neuen zusammensetzen.

Die Ergebnisse der Textfunktionen sind entweder Zeichenketten, die Textvariablen zugewiesen oder in Textausdrücken weiterverarbeitet werden können, oder numerische Werte, die sich in numerischen Ausdrücken verarbeiten oder numerischen Variablen zuordnen lassen.

Folgende Textfunktionen stehen zur Verfügung:

ASC	Umwandlung eines Zeichens in den zugehörigen ASCII-Code.
CHR\$	Umwandlung des ASCII-Codes in das zugehörige Zeichen.
LEFT\$	Entnimmt einer Zeichenfolge eine spezifizierte Zeichenanzahl von links.
RIGHT\$	Entnimmt einer Zeichenfolge eine spezifizierte Zeichenanzahl von rechts.
MID\$	Entnimmt einer Zeichenfolge eine spezifizierte Zeichenanzahl aus der Mitte.
STR\$	Wandelt den Wert eines numerischen Ausdrucks in eine Zeichenfolge um.
VAL	Wandelt eine als Zeichenfolge eingegebene Zahl in ihren numerischen Wert um.
LEN	Berechnet die Anzahl der Zeichen eines Textausdruckes.

## 2.2.6 PROGRAMMIEREN IN BASIC

Der Computer verwendet die weit verbreitete Programmiersprache **BASIC** (Beginner's All-Purpose Symbolic Instruction Code). Diese Programmiersprache wurde 1960 in Dartmouth College entwickelt und hat sich bis heute insbesondere bei den Mikrocomputern durchgesetzt.

**BASIC** ist im Gegensatz zu den anderen Programmiersprachen einfach und leicht verständlich.

**BASIC** gestattet den Dialogbetrieb mit dem Rechner und ist dabei so flexibel, daß ein bestehendes Programm ohne großen Aufwand geändert werden kann.

Abgesehen von einigen Abweichungen von **BASIC**-Version zu **BASIC**-Version ist diese Programmiersprache maschinenunabhängig.

### 2.2.6.1 BASIC - Übersicht

Wie jede natürliche Sprache hat auch die formale Computersprache **BASIC** eine Grammatik und einen zugehörigen Zeichensatz.

Letzterer setzt sich aus folgenden Zeichen zusammen:

Buchstaben: A, B, . . . , Z

Zahlen: 0, 1, . . . , 9

Sonderzeichen: ( ) . ; : + - \* / \$ & % # @ ! ? < > = ^

Zur Vermeidung von Verwechslungen unterscheidet man zwischen Ø (Null) und dem Buchstaben O.

**BASIC** kennt folgende Sprachelemente:

- numerische Konstanten
- Textkonstanten
- numerische Variablen
- Textvariablen
- dimensionierte oder Feldvariablen
- **BASIC**-Schlüsselwörter für
  - Standardfunktionen
  - Zuweisungen
  - Eingaben
  - Ausgaben
  - Steuerungen
  - Kommandos
- Operationszeichen (+, -, \*, /, ^, >, <, =, >=, <=, <>)

Diese Sprachelemente werden zu Programmsätzen zusammengefügt, die vom Rechner der angegebenen Reihenfolge nach abgeleitet werden.

### 2.2.6.2 Programmerstellung

Die Programmerstellung gliedert sich im wesentlichen in drei Schritte:

#### PROBLEMANALYSE

#### UMSETZUNG IN DIE PROGRAMMIERSPRACHE UND EINGABE IN DEN COMPUTER

#### TESTEN DES PROGRAMMS

##### **Problemanalyse:**

Das gestellte Problem muß zunächst analysiert und dann so aufbereitet werden, daß sich die einzelnen Teilprogramme leicht und übersichtlich programmieren lassen.

##### **Umsetzung in die Programmiersprache und Eingabe in den Computer:**

Die in die einzelnen Schritte zerlegte Aufgabenstellung wird in eine Programmiersprache (hier BASIC) übertragen und in den Computer eingegeben.

##### **Testen des Programms:**

Nur sehr selten wird das umgesetzte und eingegebene Programm auf Anhieb fehlerfrei arbeiten. Dabei können die verschiedenartigsten Fehler auftreten:

- Schreibfehler
- falsche Sprachelemente (SYNTAX-Fehler)
- falscher logischer Aufbau
- falsche Programmstruktur (fehlerhafte Problemanalyse)

Der Computer bietet einige Möglichkeiten, auftretende Fehler zu finden und zu beheben (siehe 2.2.6.8 Fehlermeldungen/Fehlersuche).

### 2.2.6.3 Programmaufbau

Bei der Übertragung der Problemanalyse in ein BASIC-Programm sind die im folgenden aufgeführten Regeln zu beachten:

- Ein vollständiges BASIC-Programm besteht aus einer Reihe von Anweisungen, die in der Reihenfolge angeordnet sein müssen, in der sie später ausgeführt werden sollen. Eine Ausnahme hierfür sind die GOTO-, GOSUB-, ON GOTO- und ON GOSUB-Anweisungen, die das Programm an eine festgelegte Stelle verzweigen können.
- Die Programmierung erfolgt zeilenweise, wobei jede Programmzeile eine oder mehrere durch Doppelpunkt getrennte Programmanweisungen enthalten kann.
- Die Anweisungen dürfen nicht länger als eine Zeile sein, d.h., sie können nicht in der nächsten Zeile fortgesetzt werden. Eine Zeile kann bis zu 79 Zeichen enthalten, wovon jeweils 16 Zeichen auf dem Display angezeigt werden. Die restlichen Zeichen einer Zeile können mit den (Cursor-)Tasten oder sichtbar gemacht werden.

**Hinweis:** BASIC-Schlüsselwörter wie LET, SIN, COS, PRINT usw. werden nach Betätigung der **ENTER**-Taste intern abgekürzt. Bei der Eingabe einer sehr langen Programmzeile kann man vorerst nur 79 Zeichen eingeben. Nach Drücken der **ENTER**-Taste werden die Schlüsselwörter abgekürzt, und man kann zusätzlich noch einige Zeichen eingeben. Die Eingabe muß wieder mit **ENTER** abgeschlossen werden.

- Jede Zeile muß mit einer positiven ganzen Zahl (der Zeilennummer) beginnen. Eine Zeilennummer darf nur einmal im Programm verwendet werden. Die Programmzeilen werden vom Rechner in aufsteigender Reihenfolge ausgeführt, wenn nicht durch Steueranweisungen (z.B. GOTO) eine andere Reihenfolge festgelegt wird. Die Numerierung muß nicht lückenlos sein, es ist sogar zweckmäßig, z.B. Zehner-schritte zu wählen, da damit die Möglichkeit geboten ist, nachträglich noch Programmzeilen einzufügen. Die Zeilennummern dürfen im Bereich von 1 bis 65279 gewählt werden.

### 2.2.6.4 Eingabe eines Programms

Die Programmeingabe erfolgt im PRO-Mode. Ein sich eventuell noch im Speicher befindliches Programm kann mit der NEW-Anweisung gelöscht werden.

**Beispiel einer Programmeingabe:**

NEW

```

10 A = 15
20 B = 3
30 C = A + B
40 PRINT C
50 END

```

Jede Programmzeile muß mit **[ENTER]** abgeschlossen werden. Der Rechner fügt dann einen Doppelpunkt zwischen die Zeilennummer und der ersten Anweisung ein.

**Beispiel:**

```
10 A = 15 [ENTER]
```

Anzeige:

```
10: A = 15
```

**Überprüfen des Programms, Editieren und Auflisten**

Ist die Programmeingabe abgeschlossen, so kann man im PRO-Mode den Inhalt der einzelnen Zeilen noch einmal überprüfen.

Betätigt man die **[↑]**-Taste, wird die vorhergehende Programmzeile angezeigt.

Betätigt man die **[↓]**-Taste, wird die folgende Programmzeile angezeigt.

Mit der LIST-Anweisung kann man genau spezifizierte Zeilen zur Anzeige bringen.

**Beispiele:**

LIST	Die erste Zeile eines Programms wird angezeigt.
------	---

LIST 100	Die Programmzeile 100 wird angezeigt.
----------	---------------------------------------

Mit der LLIST-Anweisung kann das Programm über den Drucker (Option CE-126P) auf Papier 'gelistet' werden.

**Beispiele:**

LLIST	Das gesamte Programm wird ausgedruckt.
-------	--

LLIST 100	Die Programmzeile 100 wird ausgedruckt.
-----------	---

LLIST 100,200	Die Programmzeilen von 100 bis 200 werden ausgedruckt.
---------------	--

LLIST 100,	Das Programm wird ab Programmzeile 100 ausgedruckt.
------------	---

LLIST ,100	Das Programm wird bis Programmzeile 100 ausgedruckt.
------------	--

**Beispiel:**

```

10 A = 15
20 B = 3
30 C = A + B
40 PRINT C
50 END

```

Durch Eingabe von LIST 30 (nur im PRO-Mode möglich) wird am Display folgendes angezeigt:

**30: C = A + B**

Die Zeile 40 kann jetzt durch Drücken der **↓**-Taste angezeigt werden.

**40: PRINT C**

### 2.2.6.5 Korrektur einer Zeile

Die Korrektur einer Zeile (nur im PRO-Mode) ist sehr einfach und kann auf zwei verschiedene Arten erfolgen:

#### 1. Überschreiben der Zeile:

Man gibt die neue Zeile mit derselben Zeilennummer ein. Die alte Version der Programmzeile wird dadurch überschrieben.

#### 2. Editieren:

Man bringt die zu editierende Zeile mit LIST zur Anzeige und drückt eine der beiden Cursor-Tasten (**◀** oder **▶**). Damit wird der Doppelpunkt zwischen der Zeilennummer und der ersten Programmanweisung unterdrückt und die Zeile zur Korrektur freigegeben. Jetzt können Sie mit Hilfe der **SHIFT** **DEL** - oder der **INS** -Taste Zeichen in der Zeile löschen oder einfügen. Auch ein Überschreiben einzelner Zeichen ist möglich.

**Beispiel:**

In dem oben angeführten Beispiel soll die Zahl 15 durch die Zahl 17 ersetzt werden. Hierzu geben Sie ein:

**LIST 10 ENTER** (Zeile 10 anzeigen)

**◀** (Beginn Änderung)

**▶ ▶ ▶** (Cursor auf die '5')

**7 ENTER** ('5' durch '7' ersetzen)

### 2.2.6.6 Löschen einer Zeile

Das Löschen einer Programmzeile erfolgt im PRO-Mode. Ein Zeile wird ersatzlos gelöscht, wenn man nur die Zeilennummer eingibt und die **ENTER**-Taste drückt.

#### Beispiel:

```
10 A = 15  
20 B = 3  
30 C = A + B  
40 PRINT C  
50 END
```

Eingabe:

```
10 ENTER  
20 ENTER
```

Listet man nun das Programm, so sieht man, daß die Zeilen 10 und 20 gelöscht wurden.

### 2.2.6.7 Programmausführung

Die Programmausführung kann nur im RUN-Mode erfolgen. Der Programmstart kann durch drei verschiedene Anweisungen erfolgen (siehe RUN/GOTO/Definable Keys).

Nach Eingabe des Startkommandos beginnt der Computer mit der Abarbeitung der einzelnen Programmzeilen. Während der Programmausführung ist auf der Anzeige das Wort "BUSY" sichtbar. Dies erlischt entweder am Programmende oder wenn eine Anzeige am Display erfolgt. Solange 'BUSY' aufleuchtet, nimmt der Computer keine Eingaben von der Tastatur an. Nach Beendigung des Programms wird am Display das 'Bereitschaftssymbol' (>) angezeigt.

Zur Unterbrechung einer Programmausführung dient die **BRK**-Taste. Am Display wird folgendes angezeigt:

BREAK IN XX wobei XX die Zeilennummer ist, die vor der Unterbrechung bearbeitet wurde.

Will man nach einer Unterbrechung das Programm fortsetzen, so genügt es, die CONT(inue)-Anweisung einzugeben:

CONT

Die Programmausführung wird fortgesetzt.

### 2.2.6.8 Fehlermeldung/Fehlersuche

Tritt während der Programmäusführung ein Fehler auf, so wird dieser erkannt und gemeldet. Die Fehlermeldung wird am Display in folgender Form angezeigt:

**ERROR Fehlercode IN Zeilennummer**

Die Liste der Fehlermeldungen und ihre Erklärung finden Sie numerisch geordnet im Anhang.

Gleichzeitig mit dem Fehlercode wird die Nummer der Zeile, in der der Fehler auftrat, angezeigt. Diese Zeile kann nun auch im RUN-Mode am Display zur Anzeige gebracht werden. Löschen Sie die Fehlermeldung mit der **C-CE**-Taste und drücken Sie die Taste **↑**. Die Zeile erscheint auf der Anzeige, und der Cursor blinkt auf dem fehlerhaften Sprachelement. Will man die Zeile korrigieren, muß man in den PRO-Mode umschalten und wie in Abschnitt 2.2.6.5 beschrieben vorgehen.

#### Beispiel:

```
10 A = 2
20 B = 10
30 C = B/A
40 PRINT C,
50 END
```

Bei der Ausführung dieses Programms erscheint:

**ERROR 1 IN 40**

Die PRINT-Liste in Zeile 40 ist unvollständig. Schreiben Sie hinter das Komma ein A. Das Programm kann dann weiterlaufen.

**40 PRINT C,A**

Das Programm ist jetzt fehlerfrei.

## 2.3 BASIC REFERENZTEIL

Das folgende Kapitel ist in drei Teile untergliedert:

- 2.3.1 KOMMANDOS** Instruktionen an den Rechner, die außerhalb eines Programms gegeben werden, um Arbeitsbedingungen, Zusatzschaltungen und Programmkontrollen zu bestimmen.
- 2.3.2 BEFEHLE** Kommando- und Befehlwörter, die gebraucht werden, um ein Programm zu erstellen.
- 2.3.3 FUNKTIONEN** Spezielle BASIC-Operatoren, um Variable umzuwandeln.

Die Kommandos, die die Funktionen serieller E/A-Anschluß und Text betreffen, werden auf den folgenden Seiten dargestellt:

Serieller E/A-Anschluß	Seite 191–207
Textfunktionen	Seite 208–210

Die Kommandos und die Befehle sind alphabetisch geordnet. Um einen Befehl oder ein Kommando schnell finden zu können, beginnt jede Eingabe auf einer neuen Seite. Der Inhalt jedes Abschnitts ist in den Tabellen auf dieser und folgenden Seiten wiedergegeben. Anhand dieser Tabellen können Sie ein Schlüsselwort schnell der entsprechenden Kategorie zuordnen und nachschlagen. Funktionen sind in vier Gruppen und in diesen alphabetisch geordnet.

Achtung: Kommandos, Befehle und Funktionen müssen immer in Großbuchstaben eingegeben werden (normaler Modus).

### KOMMANDOS

Programmkontrolle		Variablenkontrolle	
CONT	NEW	CLEAR*	
GOTO*	RUN	DIM*	
Cassettenkontrolle		Winkelmodus	
CLOAD	INPUT#*	DEGREE*	RADIAN*
CLOAD?	MERGE	GRAD*	
CSAVE	PRINT#*		

**Debug-Kontrolle**

**LIST** TROFF\*  
**LLIST** TRON\*

Andere  
BEEP\*  
PASS  
RANDOM\*

USING\*  
WAIT\*

**Text**

**BASIC**  
**TEXT**

Serieller E/A-Anschluß

CLOSE\*  
CONSOLE\*  
INPUT#1\*  
LLIST

LOAD  
LPRINT\*  
OPEN\*  
OPEN\$  
PRINT#1\*  
SAVE

- \* Diese Kommandos sind auch BASIC-Befehle. Ihre Wirkung als Kommando ist identisch mit ihrer Wirkung als Befehl. Sie sind beschrieben in dem Abschnitt BEFEHLE.

**BEFEHLE****Kontrollen und Verzweigungen****Eingabe und Ausgabe**

CHAIN  
END  
FOR ... TO ... STEP  
GOSUB  
GOTO  
IF ... THEN  
NEXT  
ON ... GOSUB  
ON ... GOTO  
RETURN  
STOP

AREAD  
CSAVE  
DATA  
INPUT  
INPUT#  
INPUT#1  
LOAD  
LPRINT  
PAUSE  
PRINT  
PRINT#  
PRINT#1  
READ  
RESTORE  
USING  
WAIT

**Zuweisung und Deklarationen**

CLEAR  
DIM  
LET

**Andere**

BEEP  
CLOSE  
CONSOLE  
DEGREE  
GRAD  
OPEN

RADIAN  
RANDOM  
REM  
TROFF  
TRON

**FUNKTIONEN****Pseudovariable**

INKEY\$	OPEN\$
MEM	PI
<b>String-Funktionen</b>	
ASC	MID\$
CHR\$	RIGHT\$
LEFT\$	STR\$
LEN	VAL

**Numerische Funktionen**

ABS	INT
ACS	IN
AHC	LOG
AHS	POL
AHT	RCP
ASN	REC
ATN	RND
COS	ROT
CUR	SGN
DEG	SIN
DMS	SQR
EXP	SQU
FACT	TAN
HCS	TEN
HSN	
HTN	

### 2.3.1 KOMMANDOS

#### CLOAD

1. CLOAD

2. CLOAD "dateiname"

Abkürzung: CLO., CLOA.

Vergleiche: CLOAD?, CSAVE, MERGE, PASS

#### Wirkung:

Mit dem CLOAD-Kommando werden Programme von der Cassette in den Rechner geladen.

#### Anwendung:

Die erste Form des CLOAD-Kommandos löscht den Speicher des Rechners und lädt das erste Programm auf der eingelegten Cassette.

Die zweite Form des Kommandos löscht den Speicher und lädt das mit "dateiname" angegebene Programm von der Cassette.

Ist der Computer im PROgramm- oder RUN-Modus, so wird das Programm von der Cassette in den Programmspeicher geladen.

#### Beispiele:

CLOAD : Lädt das erste Programm von der Cassette.

CLOAD "PRO3" : Sucht auf der eingelegten Cassette das Programm PRO3 und lädt es ein.

#### Bemerkungen:

1. Wird die angegebene Datei nicht gefunden, sucht der Rechner weiter nach der Datei, selbst wenn die Cassette abgelaufen ist. Unterbrechen Sie in diesem Fall die Suche mit der **BRK**-Taste. Das bezieht sich auch auf die Kommandos MERGE, CHAIN, CLOAD? und INPUT#, die später beschrieben werden.
  2. Tritt während der Ausführung der Kommandos CLOAD oder CHAIN (wird später beschrieben) ein Fehler auf, so ist das im Speicher vorhandene Programm defekt.
- \* Während des Ladevorgangs wird in der rechten Ecke ein Symbol "\*" angezeigt. Dieses Symbol verschwindet, wenn das Laden beendet ist. Wird ein Dateiname gesucht, erscheint dieses Symbol nicht, bevor der Ladevorgang beginnt.

**CLOAD?**

1. CLOAD?
2. CLOAD? "dateiname"

Abkürzung: CLO.?, CLOA.?

Vergleiche: CLOAD, CSAVE, MERGE, PASS

**Wirkung:**

Mit dem Kommando CLOAD? vergleichen Sie das Programm im Speicher des Rechners mit einem auf Cassette gespeicherten Programm.

**Anwendung:**

Um zu prüfen, ob ein Programm richtig gespeichert wurde, spulen Sie die Cassette an den Anfang zurück und geben das Kommando CLOAD? ein.

Die erste Form des CLOAD?-Kommados vergleicht den Speicherinhalt mit dem ersten auf der Cassette gefundenen Programm.

Die zweite Form des Kommandos CLOAD? sucht nach dem mit "dateiname" angegebenen Programm und vergleicht es dann mit dem Speicherinhalt.

**Beispiele:**

CLOAD? Vergleicht den Speicherinhalt mit dem ersten auf der Cassette gefundenen Programm.

CLOAD? "PRO3" Sucht auf der eingelegten Cassette nach dem Programm PRO3 und vergleicht es dann mit dem Speicherinhalt.

\* Das Symbol "\*" erscheint ganz rechts auf dem Display, während das Programm überprüft wird. Am Ende des Prüfvorgangs verschwindet das Sternchen, und es erscheint das Bereitschaftssymbol.

## CONT

### 1. CONT

Abkürzung: C., CO., CON.

Vergleiche: RUN, STOP-Befehl

#### Wirkung:

Mit dem CONT-Kommando setzen Sie die Ausführung eines unterbrochenen Programms fort.

#### Anwendung:

Wurde die Ausführung eines Programms mit dem Befehl STOP oder mit der Taste **BRK** unterbrochen, so kann es mit dem Kommando CONT fortgesetzt werden.

CONT können Sie auch benutzen, wenn die Ausführung des Programms zeitweise beispielsweise durch Kommandos wie PRINT unterbrochen wurde.

#### Beispiel:

CONT      Setzt die Ausführung eines unterbrochenen Programms fort.

## CSAVE

1. CSAVE
2. CSAVE "dateiname"
3. CSAVE, "passwort"
4. CSAVE "dateiname", "passwort"

Abkürzung: CS., CSA., CSAV.

Vergleiche: CLOAD, CLOAD?, MERGE, PASS

### Wirkung:

Mit diesem Kommando wird ein Programm aus dem Speicher des Rechners auf der Cassette abgespeichert.

### Anwendung:

Die erste Form des Kommandos CSAVE speichert das Programm im Speicher des Rechners ohne speziellen Dateinamen auf der Cassette ab.

Die zweite Form des Kommandos weist dem Programm vor Abspeicherung den angegebenen Dateinamen zu und speichert es unter diesem Namen auf der Cassette ab.

Die dritte Form des Kommandos speichert das Programm ohne Dateinamen, aber mit einem Passwort auf der Cassette ab. Programme, die mit einem Passwort geschützt sind, können von jedem geladen und ausgeführt werden. Gelistet oder verändert werden kann es aber nur dann, wenn das korrekte Passwort eingegeben wird (näheres unter der Beschreibung des PASS-Kommandos).

Mit der vierten Form des CSAVE-Kommandos wird den Programm sowohl ein Dateiname als auch ein Passwort zugewiesen.

### Beispiel:

CSAVE "PRO3", "GEHEIM"      Speichert das präsente Programm unter dem Dateinamen PRO3 mit dem Passwort GEHEIM auf Cassette ab.

## GOTO

### 1. GOTO ausdruck

Abkürzung: G., GO., GOT.

Vergleiche: RUN

#### Wirkung:

Mit dem GOTO-Kommando wird die Ausführung eines Programms begonnen:

#### Anwendung:

Das GOTO-Kommando kann genauso wie das RUN-Kommando benutzt werden. Die Ausführung eines Programms wird von der durch ausdruck angegebenen Zeilennummer gestartet.

GOTO unterscheidet sich in sechs Punkten vom RUN-Kommando:

1. Der Wert des WAIT-Intervalles wird nicht zurückgesetzt.
2. Wurde die Anzeige durch einen USING-Befehl formatiert, wird sie nicht gelöscht.
3. Werte von Variablen und Feldern (Arrays) bleiben erhalten.
4. PRINT=LPRINT-Status wird nicht zurückgesetzt.
5. Der READ-Zeiger wird nicht zurückgesetzt.
6. Der serielle E/A-Anschluß ist nicht geschlossen.

#### Beispiel:

**GOTO 100**      Die Ausführung eines Programmes wird bei Zeile 100 gestartet.

## LIST

1. LIST
2. LIST zeilennummer
3. LIST "label"

Abkürzung: L., LI., LIS.

Vergleiche: LLIST

### Wirkung:

Mit dem LIST-Kommando wird ein Programm zur Anzeige gebracht.

### Anwendung:

Das LIST-Kommando kann nur im PROgramm-Modus benutzt werden.

- \* Mit LIST wird das Programm von der ersten Zeile angezeigt, so weit, bis das Display voll ist.
- \* Mit LIST zeilennummer wird das Programm von der angegebenen Zeilennummer angezeigt, so weit, bis das Display voll ist.  
Gibt es die angegebenen Zeilennummer innerhalb des Programms nicht, so wird das Programm ab der Zeile angezeigt, deren Zeilennummer der angegebenen am nächsten kommt.
- \* Mit LIST "label" wird das Programm ab der mit diesem Etikett versehenen Zeile angezeigt, so weit, bis das Display voll ist.
- \* Sind mit dem MERGE-Kommando mehrere Programme in den Speicher geladen, so wird mit dem LIST-Kommando das letzte eingeladene Programm gelistet.
- \* Wird das Kommando in der Form LIST "label" benutzt und das angegebene Label im zuletzt geladenen Programm nicht gefunden, so sucht das Kommando das Label in den anderen Programmen im Speicher. Die dazugehörige Zeile wird dann gelistet.

Bei einem durch ein Passwort geschützten Programm wird das LIST-Kommando nicht ausgeführt.

### Beispiel:

LIST 100      Die Zeile 100 wird angezeigt.

## LLIST

1. LLIST
2. LLIST ausdruck
3. LLIST ausdruck 1, ausdruck 2
4. LLIST ausdruck,
5. LLIST ,ausdruck
6. LLIST "label"

Abkürzung: LL., LLI., LLIS.

Vergleiche: LIST

### Wirkung:

Mit dem LLIST-Kommando kann ein Programm auf dem Drucker ausgegeben werden.

Die Ausgabe kann durch das OPEN-Kommando geöffnet und durch das CLOSE-Kommando geschlossen werden.

### Anwendung:

\* Wenn der serielle E/A-Anschluß durch das OPEN-Kommando geöffnet würde, kann durch das LLIST-Kommando das gewünschte Programm auf dem an den E/A-Anschluß angeschlossenen Terminal ausgegeben werden. Um diese Ausgabe abzubrechen, benutzen Sie das CLOSE-Kommando.

Das LLIST-Komando kann im PROgramm-Modus und im RUN-Modus benutzt werden.

Die erste Form druckt das gesamte Programm in Speicher aus. Die zweite Form druckt nur die mit ausdruck angegebene Zeile aus. Die dritte Form des Kommandos druckt das Programm von der durch den ausdruck1 angegebenen Zeile oder der nächstgrößeren bis zu der durch den ausdruck2 angegebenen Zeile oder der nächstgrößeren aus. Zwischen den beiden angegebenen Zeilen müssen mindestens zwei Zeilen liegen. Die vierte Form listet alle Programme ab der durch ausdruck eingegebenen Zeile bis zum Ende. Die fünfte Form des Kommandos LLIST druckt die Programmzeilen bis zur eingegebenen Zeile (inklusive) aus.

Wenn mit dem MERGE-Kommando mehrere Programme eingeladen wurden, so arbeitet das LLIST-Kommando mit dem zuletzt geladenen Programm. Um ein zuvor geladenes Programm aufzurufen, benutzen Sie das Kommando LLIST "label".

Ein gesetztes Passwort unterdrückt das LLIST-Kommando.

\* Zum Ausdrucken eines Programms auf dem Drucker CE-140P die Anzahl der Zeichen pro Zeile auf 24 oder mehr einstellen. (Siehe das CONSOLE-Kom-

mando.) Wenn die Anzahl der Zeichen pro Zeile auf weniger als 24 eingestellt ist und der Befehl LLIST ausgeführt wird, tritt ein Fehler (ERROR 3) auf.

- \* Beim Ausdrucken von Programmen auf dem Drucker CE-140P wird "√" als SQR und "π" als PI ausgedruckt.

Beispiel:

LLIST 100, 200

Die Programmzeilen zwischen 100 und 200 werden ausgedruckt.

**MERGE**

1. MERGE

2. MERGE "dateiname"

Abkürzung: MER., MERG.

Vergleiche: CLOAD

**Wirkung:**

Mit dem MERGE-Kommando können Programme von der Cassette an bereits im Speicher vorhandene angeknüpft werden.

**Anwendung:**

Mit dem MERGE-Kommando wird das bereits im Speicher geladene Programm aufbewahrt und ein weiteres Programm von der Cassette in den Speicher des Rechners geladen. Auf diese Weise können verschiedene Programme gleichzeitig in den Computer geladen werden.

**Beispiele:**

Nehmen wir an, wir haben drei Programme mit den Dateinamen PRO1, PRO2 und PRO3 auf Cassette abgespeichert. Jetzt kann das Programm PRO1 mit dem Kommando CLOAD eingeladen werden, die Programme PRO2 und PRO3 mit dem MERGE-Kommando. Die Abbildung auf dieser Seite verdeutlicht, wie die Programme gespeichert werden.

## Cassette

PRO-1	/ / / /	PRO-2	/ / / /	PRO-3	/ / / /
dateiname Programm		dateiname Programm		dateiname Programm	

CLOAD "PRO-1"

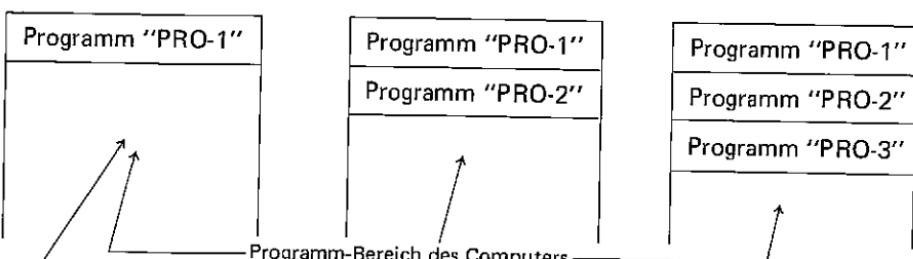
ENTER

MERGE "PRO-2"

ENTER

MERGE "PRO-3"

ENTER



- Benutzen Sie das CLOAD-Kommando, um das erste Programm in den Rechner zu laden.

Programme, die mit dem MERGE-Kommando geladen wurden, werden wie im o. g. Beispiel gespeichert.

- \* Ist die erste Zeilennummer des mit dem MERGE-Kommando geladenen Programms größer als die letzte Zeilennummer des vorher geladenen Programms, werden die beiden Programme im folgenden wie eins behandelt.
- \* Ist die erste Zeilennummer des mit dem MERGE-Kommando geladenen Programms kleiner als die letzte Zeilennummer des vorher geladenen Programms, werden die beiden Programme als zwei verschiedene behandelt.  
In dem obigen Beispiel sind die Zeilennummern für die ausgewählten Programme PRO1, PRO2 und PRO3 10–200, 50–150 und 160–300. Die Programme PRO1 und PRO2 werden als zwei verschiedene Programme behandelt. PRO2 und PRO3 werden bei den Zeilennummern 50–300 als ein Programm behandelt.
- \* Wenn Sie Programme mit dem MERGE-Kommando laden, kann es zwei oder mehrere Programme mit den gleichen Zeilennummern geben. In diesem Fall beziehen sich die Kommandos RUN oder GOTO in der Form RUN ausdruck bzw. GOTO ausdruck auf das letzte, mit MERGE geladene Programm. Dann können die zuvor geladenen Programme nicht mehr ausgeführt werden.

Geben Sie deshalb immer ein Label am Beginn eines Programms an und lassen Sie es anschließend ausführen.

Beachten Sie, daß in jedem Fall nur das zuletzt mit MERGE geladene Programm editiert werden kann, wenn Sie das MERGE-Kommando geben. Alle zuvor mit MERGE geladenen Programme können nicht editiert werden. Geben Sie deshalb immer das Label eines Programms an, ehe Sie mit dem MERGE-Kommando das nächste Programm laden.

## VERKNÜPFUNG PASSWORT-GESCHÜTZTER PROGRAMME

Wollen Sie ein Passwort-geschütztes Programm mit dem MERGE-Kommando laden, so müssen wir zwei Fälle unterscheiden:

1. Das Programm im Speicher ist geschützt.
2. Das Programm im Speicher ist nicht geschützt.

Im ersten Fall kann kein Passwort-geschütztes Programm mit MERGE geladen werden. Im zweiten Fall werden alle Programme im Speicher durch das neu geladene, geschützte Programm ebenfalls zu Passwort-geschützten Programmen.

Sind im Speicher geschützte Programme und wird ein ungeschütztes Programm mit MERGE geladen, so wird dieses neu geladene Programm ebenfalls zu einem Passwort-geschützten Programm.

#### AUSFÜHRUNG MIT MERGE GELADENER PROGRAMME

Die Abbildung zeigt die Speicherorganisation, nachdem PRO1 geladen wurde. Anschließend wurde mit dem MERGE-Kommando PRO2 und PRO3 geladen. Wenn Sie RUN oder GOTO (RUN ausdruck, GOTO ausdruck) benutzen, um ein Programm zu starten, so wird immer PRO3 ausgeführt.

Der Rechner kann mit Hilfe von Labels verschiedene Programme auf dem Speicher ablaufen lassen. Dazu müssen die Programme PRO1, PRO2 und PRO3 im Speicher unterteilt werden. Die Tasten A, B und C sind dabei die Labels für diese Programme. Der Rechner kann nun über die Tasten A, B oder C das Programm PRO1, PRO2 oder PRO3 starten. Das Programm PRO1 besteht aus den Zeilen 1000 bis 1009, PRO2 aus den Zeilen 2000 bis 2009 und PRO3 aus den Zeilen 3000 bis 3009.

Wenn der Rechner auf diese Weise ein Label sucht und das gewünschte Label in Programm PRO3 auftaucht, können die Programme PRO1 und PRO2 nicht ausgeführt werden.

Weiterhin gilt: Wird das LIST-Kommando mit Label-Angabe in der Form LLIST "A", "C" **ENTER** eingegeben, sucht der Rechner zuerst im Programm PRO3. Wenn "C" zuerst erscheint, erhalten Sie eine Fehlermeldung.

**NEW**

## 1. NEW

**Wirkung:**

Das NEW-Kommando löscht Programme bzw. Daten im Speicher.

**Anwendung:**

Wird es im PROgramm-Modus benutzt, so werden alle Programme und Daten gelöscht. (Mit Passwort geschützte Programme können nicht gelöscht werden.)

Das NEW-Kommando ist im RUN-Modus nicht definiert. Sie erhalten nach Eingabe des NEW-Kommandos die Fehlermeldung ERROR 9.

**Beispiel:**

NEW            Löscht den entsprechenden Speicherbereich.

## PASS

### 1. PASS "zeichenkette"

Abkürzung: PA., PAS.

Vergleiche: CSAVE, CLOAD, NEW

#### Wirkung:

Mit dem Kommando PASS können Passwörter gesetzt und gelöscht werden.

#### Anwendung:

Mit einem Passwort schützen Sie Ihr Programm vor Einsichtnahme oder Modifikation durch andere Anwender. Ein Passwort besteht aus einer Zeichenkette bis zu sieben Zeichen. Die erlaubten Zeichen für das Passwort sind alle Buchstaben des Alphabets und folgende Sonderzeichen:

!	#	\$	%	&	(	)	*	*	*	*	*	*	*
+	-	/	.	,	:	;	<						
=	>	?	@	^	√	%							

sowie das Wurzelzeichen und Pi.

Wurde das Kommando PASS gegeben, so ist das Programm im Speicher geschützt. Das bedeutet, daß das Programm nicht mehr editiert werden kann. Es kann nicht auf Cassette abgespeichert oder mit LIST oder LLIST ausgegeben werden. Weiterhin können weder Zeilen hinzugefügt noch gelöscht werden. Die einzige Möglichkeit, das Passwort auszuschalten, ist ein weiteres PASS-Kommando mit dem gleichen Passwort einzugeben.

Achtung: Wenn Sie ein Passwort mit mehr als 7 Zeichen eingeben, werden nur die ersten 7 Zeichen als gültig erkannt und dem Programm zugewiesen.

Geben Sie **ENTER** ein, nachdem Sie das Passwort bestimmt haben.

Sie erhalten eine Fehlermeldung, wenn Sie nach der Eingabe des Passwortes weitere Zeichen oder Symbole eingeben. In einem solchen Fall akzeptiert der Rechner das Passwort nicht. Es erfolgt eine Fehlermeldung nach Eingabe von PASS "ABCDEFG": A = 123 **ENTER**.

#### Beispiel:

PASS "GEHEIM"

Setzt für alle Programme im Speicher das Passwort "GEHEIM".

\* Wenn das Passwort gesetzt wurde und die RAM-Karte entnommen wird, ist das in der RAM-Karte gespeicherte Programm geschützt.

## RUN

1. RUN
2. RUN zeilennummer

Abkürzung: R., RU.

Vergleiche: GOTO, MERGE

### Wirkung:

Mit dem RUN-Kommando wird das Programm im Speicher gestartet.

### Anwendung:

Die erste Form des Kommandos startet das Programm im Speicher mit der Zeile, die die niedrigste Zeilennummer hat.

Die zweite Form startet das Programm mit der angegebenen Zeilennummer. Die Kommandos RUN und GOTO unterscheiden sich in sechs Punkten.

\* Wenn mehrere Programme mit dem MERGE-Kommando geladen werden, wird das zuletzt geladene Programm mit dem Kommando RUN oder mit dem Kommando RUN ausdruck ausgeführt.

Für das Kommando RUN gelten folgende Merkmale:

1. Der Wert des WAIT-Intervalls wird zurückgesetzt.
2. Eine durch einen USING-Befehl erzeugte Formatierung der Anzeige wird gelöscht.
3. Variable und Felder (Arrays) werden gelöscht.
4. PRINT=PRINT-Status wird gesetzt.
5. Der READ-Zeiger wird auf den Anfang der ersten DATA-Zeile gesetzt.
6. Schließt den seriellen E/A-Anschluß (serieller Anschluß).

Die Ausführung eines Programms mit GOTO und mit der **DEF**-Taste sind gleich. Alle drei Formen des Programmstarts löschen FOR/NEXT- und GOTO-Gruppen.

### Beispiel:

RUN 100

Startet ein Programm mit der Zeile 100.

## 2.3.2 BEFEHLE

### AREAD

#### 1. AREAD variablenname

Abkürzung: A., AR., ARE., AREA.

Vergleiche: Befehl INPUT und die Beschreibung der **[DEF]**-Taste.

#### Anwendung:

Der AREAD-Befehl wird benutzt, um eine einzelne Variable in ein Programm einzulesen, welches mit der **[DEF]**-Taste gestartet wurde.

#### Wirkung:

Wenn einem Programm ein Label (ein "Name") in Form eines Buchstabens zugewiesen wurde, so daß es mit der **[DEF]**-Taste gestartet werden kann, kann der AREAD-Befehl benutzt werden, um einen einzelnen Startwert einzugeben, ohne den INPUT-Befehl zu setzen. Der Befehl AREAD muß in der ersten Zeile nach der Zeile mit dem Label gesetzt sein. Taucht er an anderer Stelle im Programm auf, so wird er ignoriert. Es können sowohl Zeichenketten (Strings) als auch numerische Werte eingegeben werden, aber immer nur ein Wert pro Programm.

Erstellen Sie Ihr Programm im PROgramm-Modus. Der AREAD-Befehl kommt dann im RUN-Modus zum Tragen. Geben Sie erst den gewünschten Wert und dann drücken Sie erst **[DEF]**, gefolgt von dem Label, das Sie gesetzt haben. Wird ein String benutzt, so muß dieser nicht in Anführungsstriche gesetzt werden.

#### Beispiele:

10 "X": AREAD N

20 PRINT N ^ 2

30 END

#### Die Eingabe

7 **[DEF]** X

gibt das Quadrat der Zahl "7", also "49" aus.

#### Wichtig:

1. Wird das Bereitschaftssymbol (>) zu Beginn der Programmausführung angezeigt, so ist die benutzte Variable gelöscht.
2. Wird zu Beginn des Programms mit dem PRINT-Befehl etwas angezeigt, so wird folgendes gespeichert:

Folgendes Programm soll ausgeführt werden:

10 "A": PRINT "ABC", "DEFG"

20 "S": AREAD A\$: PRINT A\$

RUN Modus

**[DEF]** A → ABC DEFG

**[DEF]** S → DEFG

\* Wird

PRINT numerischer ausdruck, numerischer ausdruck, numerischer ausdruck,  
numerischer ausdruck

oder

PRINT string, string, string, string

ausgegeben, so wird der zuletzt angezeigte Ausdruck gespeichert.

\* Wird

PRINT numerischer ausdruck; numerischer ausdruck; numerischer ausdruck; . . .

ausgegeben, so wird der erste angezeigte Ausdruck gespeichert.

\* Wird

PRINT string; string; string; . . .

ausgegeben, so wird der zuletzt ausgegebene Ausdruck gespeichert.

**BEEP****1. BEEP ausdruck**

Abkürzung: B., BE., BEE.

**Wirkung:**

Mit dem BEEP-Befehl wird ein Ton erzeugt.

**Anwendung:**

Der Computer erzeugt auf den BEEP-Befehl hin einen oder mehrere 4kHz-Töne. Die Anzahl der Töne wird mit "ausdruck" angegeben (positive Zahlen kleiner als 9.999999999 E 99). Nur der ganzzahlige Anteil des angegebenen Ausdrucks wird im BEEP-Befehl verwertet.

BEEP kann auch mit numerischen Variablen arbeiten oder als Kommando benutzt werden. In diesem Falle werden die Töne direkt nach dem **ENTER** erzeugt.

**Beispiele:**

10 A = 5 : B\$ = "9"

Erzeugt 3 Töne.

20 BEEP 3

Erzeugt 5 Töne.

30 BEEP A

Erzeugt 4 Töne.

40 BEEP (A+4)/2

Nicht erlaubt, ERROR 9 wird ausgegeben.

50 BEEP B\$

Es wird kein Ton erzeugt, aber auch keine Fehlermeldung

60 BEEP -4

ausgegeben.

## CHAIN

1. CHAIN
2. CHAIN ausdruck
3. CHAIN "dateiname"
4. CHAIN "dateiname", ausdruck

Abkürzung: CHA., CHAI.

Vergleiche: CLOAD, CSAVE, RUN

### Wirkung:

Mit dem CHAIN-Befehl wird ein auf Cassette gespeichertes Programm ausgeführt. Er kann nur in Verbindung mit den Optionen CE-126P und CE-152 benutzt werden.

### Anwendung:

Um den CHAIN-Befehl benutzen zu können, muß mindestens ein Programm auf Cassette abgespeichert sein. Dieses Programm wird dann mit dem CHAIN-Befehl geladen und ausgeführt.

Die erste Form des Befehls lädt das erste gefundene Programm von Cassette und beginnt die Ausführung mit der niedrigsten Zeile dieses Programms. Der Effekt ist der gleiche wie mit dem CLOAD- und dem RUN-Kommando im RUN-Modus.

Die zweite Form lädt das erste gefundene Programm und beginnt die Ausführung mit der durch ausdruck angegebenen Zeile.

Die dritte Form von CHAIN sucht das mit dateiname angegebene Programm, lädt es und führt es aus, beginnend mit der kleinsten Zeilennummer dieses Programms.

Die vierte Form des CHAIN-Befehls sucht das mit dateiname bezeichnete Programm, lädt es von Cassette und startet die Ausführung mit der in ausdruck angegebenen Zeile.

### Beispiele:

10 CHAIN	Lädt das nächste Programm von Cassette und beginnt die Ausführung mit der niedrigsten Zeilennummer.
20 CHAIN "PRO2", 480	Sucht auf der Cassette das Programm PRO2, lädt es und startet es mit Zeile 480.

Nehmen Sie z.B. einmal an, daß Sie drei Programme mit dem Namen PRO1, PRO2 und PRO3 auf Cassette gespeichert haben. Alle Programme enden mit einem CHAIN-Befehl.

Während der Programmausführung wird der Rechner, wenn er auf einen CHAIN-Befehl trifft, die folgende Sektion in den Speicher rufen und sie ausführen. Auf diese Weise werden schließlich alle Sektionen abgearbeitet.

### "PRO-1"

```
10:      ↗
20:
.
.
Ausführung
```

Cassettenrecorder  
("↓" markiert die Position des  
Cassetten-Lese-Kopfes.)

```
400: CHAIN 400, 410, 420
```

Dateiname "PRO-2"	Dateiname "PRO-3"
----------------------	----------------------

### "PRO-2"

```
410:
.
.
Ausführung
```

```
700: CHAIN 700, 710, 720
```

Dateiname "PRO-2"	Dateiname "PRO-3"
----------------------	----------------------

### "PRO-3"

```
710:
.
.
Ausführung
```

```
990: END
```

Dateiname "PRO-2"	Dateiname "PRO-3"
----------------------	----------------------

```
990: END
```

Dateiname "PRO-2"	Dateiname "PRO-3"
----------------------	----------------------

Achtung: Wenn ein Programm, das ein CHAIN-Kommando enthält, vom Band durch ein MERGE-Kommando geladen werden soll, überprüfen Sie zuvor das CHAIN-Kommando auf seine Richtigkeit.

CLEAR

## 1. CLEAR

Akkürzung: CL., CLE., CLEA.

Vergleiche: DIM

### **Wirkung:**

Mit dem CLEAR-Befehl werden alle in einem Programm benutzten Variablen gelöscht und alle gesetzten Variablen auf Null bzw. auf leer gesetzt.

### Anwendung:

Der **CLEAR**-Befehl setzt Speicherraum frei, der gebraucht wurde, um Werte von Variablen zu speichern. Dieser Befehl ist sehr hilfreich, wenn der Speicherbereich eingeschränkt ist und Variablen des ersten Teils eines Programms im zweiten Teil nicht mehr gebraucht werden. Wird **CLEAR** an den Anfang eines Programms gesetzt, so werden Speicherbereiche, die von den Variablen anderer Programme im Rechner belegt sind, frei gemacht.

CLEAR setzt die Speicherbereiche, die durch die Variablen A-Z, A\$-Z\$ oder A(1)-A(26) (ohne DIMensionierung) nicht frei, wenn diese fest vorgegeben wurden. CLEAR setzt diese Variablen auf Null (numerische) bzw. auf leer (String).

### **Beispiele:**

10 A=5: DIM C(5)

**20 CLEAR** Setzt den C( ) zugewiesenen Speicher frei und A gleich Null.

## DATA

### 1. DATA ausdruck, ausdruck, ausdruck, ausdruck . . .

Abkürzung: DA., DAT.

Vergleiche: READ, RESTORE

#### Wirkung:

Der DATA-Befehl stellt für den READ-Befehl die benötigten Daten bereit.

#### Anwendung:

Soll ein Feld (Array) mit Daten gefüllt werden, empfiehlt es sich, diese Daten in DATA-Befehlen bereit zu stellen und diese Daten dann mit einer FOR/NEXT-Schleife und einem READ-Befehl in das Feld einzulesen. Der erste READ-Befehl liest den ersten Wert aus der ersten DATA-Zeile, der zweite READ-Befehl den zweiten Wert und so fort. Mit jedem READ-Befehl wird ein Zeiger auf den nächsten Wert gesetzt. Sind die Daten einer DATA-Zeile erschöpft, so wird der Zeiger an den Anfang der nächsten DATA-Zeile gesetzt.

DATA-Befehle haben keinen Einfluß auf den Ablauf des Programms, so daß sie an jeder Stelle des Programms eingefügt werden können. Viele Programmierer setzen DATA-Zeilen gerne hinter die Zeile, in der der dazugehörige READ-Befehl steht. Andere setzen alle DATA-Zeilen an den Anfang oder an das Ende eines Programmes. Wenn nötig, kann der Zeiger mit dem RESTORE-Befehl an den Anfang einer bestimmten DATA-Zeile gesetzt werden. So ist es auch möglich, eine DATA-Zeile mehrmals zu benutzen.

#### Beispiele:

10: DIM B(10)

Initialisierung eines Feldes

20: WAIT 128

30: FOR I=1 TO 10

Läßt die Werte aus den DATA-Zeilen in das Feld

40: READ B(I)

B( ). B(1) wird gleich 10, B(2) gleich 20, B(3)

50: PRINT B(I)

gleich 30 gesetzt, usw.

60: NEXT I

70: DATA 10,20,30,40,50,60

80: DATA 70,80,90,100

90: END

## DEGREE

### 1. DEGREE

Abkürzung: DE., DEG., DEGR., DEGRE.

Vergleiche: GRAD, RADIAN

#### Wirkung:

Mit dem DEGREE-Befehl werden die Winkelwerte auf Alt-Grad umgeschaltet.

#### Anwendung:

Der Computer kann drei Formen der Winkeldarstellung verarbeiten:

1. Alt-Grad (dezimal, sexagesimal)
2. Radial-Werte (Bogenmaß)
3. Neu-Grad

Diese Formen werden für die Darstellung der Werte der Argumente von SINus-, COSinus- und TANgens-Funktionen und der Ergebnisse der Umkehrfunktionen ArcusSINus, ArcusCoSinus und ArcusTaNgens benutzt. Mit der DMS- und der DEG-Funktion kann ein Dezimalwinkel in Grad, Minute und Sekunde (sexagesimal) umgeformt werden bzw. umgekehrt.

#### Beispiel:

10: DEGREE

20: X = ASN 1                           X hat jetzt den Wert 90, dem Arcussinus von 1.

30: PRINT X

**DIM****1. DIM dim liste****dim liste**

**dimensionierung** (oder **größe**) und **list** mit **dimensionierung**, **dim liste** (A1.11.45) zusammengelegt

**dimensionierung****numerische dim** oder**string dim****numerischer name (größe)****string dim****string name (größe)** oder**string name (größe) \* länge****numerischer name****erlaubte numerische variable****string name****erlaubte string variable**

**größe** (nur vorgegeben) oder **größe** mit **oder** und **zwei Zeichen** oder **zwei Zeichen\$** oder **list** mit **dimensionierung** oder **list** mit **größe** (A1.11.45) bzw. **größe** mit **zahl der elemente** (A1.11.45)

**größe****länge eines strings im array**

Abkürzung: D., DI.

**Wirkung:**

Der DIM-Befehl wird benutzt, um Speicher für numerische oder String-Felder frei zu halten.

**Anwendung:**

Außer für Felder der Form A( ), A\$( ) oder Zwei-Zeichen( ) und Zwei-Zeichen\$( ) muß jedes andere Feld mit einem DIM-Befehl initialisiert werden, um den nötigen Speicherraum bereit zu halten.

Ein Feld kann maximal 2 Dimensionen haben. Die maximale Länge einer Dimension ist 255. Zu der angegebenen Zahl addiert sich ein "nulltes" Element, so daß die Anzahl der Elemente einer Dimension immer um eins größer ist als angegeben, z.B. hat das Feld DIM B(3) die Elemente B(0), B(1), B(2) und B(3). In zweidimensionalen Feldern haben beide Dimensionen ein "Null"-Element.

In String-Feldern kann zusätzlich zur Anzahl der Elemente die Länge der einzelnen Strings vorgegeben werden, z.B. reserviert DIM B\$(3) \* 12 Speicherraum für vier String-Elemente mit je 12 Zeichen Länge. Wird die Länge der Strings nicht vorgegeben, so wird eine maximale Länge von 16 Zeichen angenommen.

Bei der Initialisierung eines Feldes werden alle Werte

- a) eines numerischen Feldes gleich Null,
- b) eines String-Feldes gleich leer

gesetzt.

Informieren Sie sich hinsichtlich der Initialisierung und Dimensionierung der Felder A( ) und A\$( ) in dem Abschnitt, in dem die Variablen erklärt werden.

**Beispiele:**

10: DIM B(10)	Reserviert Specherraum für ein numerisches Feld mit 11 Elementen.
20: DIM C\$(4,4) * 10	Reserviert Specherraum für ein zweidimensionales String-Feld mit 5 Spalten und 5 Reihen. Jeder String soll maximal 10 Zeichen lang sein.

**END****1. END**

Abkürzung: E., EN.

**Wirkung:****Der END-Befehl signalisiert das Ende eines Programms.****Anwendung:**

Sind mehrere Programme geladen, so ist es nötig, den einzelnen Programmen Endmarken zu geben, damit bei der Ausführung eines Programms der Rechner nicht in ein anderes hineinläuft. Der Befehl END setzt diese Endmarke.

Wenn der serielle E/A-Schaltkreis durch das Kommando OPEN geöffnet wurde, wird der Schaltkreis geschlossen. (Siehe das Kommando CLOSE.)

**Beispiel:**

10: PRINT "HALLO"  
20: END  
30: PRINT "TSCHUESS"  
40: END

Wenn Sie das Kommando RUN 10 geben, so gibt der Rechner das Wort HALLO aus, aber nicht das Wort TSCHUESS. Dieses wird auf das Kommando RUN 30 ausgegeben.

## FOR . . . TO . . . STEP

1. FOR numerische variable = ausdruck1 TO ausdruck2
2. FOR numerische variable = ausdruck1 TO ausdruck2  
STEP ausdruck3

Abkürzung: F., FO. ; STE.

Vergleiche: NEXT

### Wirkung:

Der FOR-Befehl wird in Verbindung mit dem NEXT-Befehl gegeben, um eine bestimmte Operation mehrmals zu wiederholen.

### Anwendung:

Der FOR- und der NEXT-Befehl schließen eine Gruppe von Befehlen ein, die wiederholt werden sollen. Wird diese Gruppe das erste Mal ausgeführt, so hat die Schleifenvariable den Wert von ausdruck1.

Erreicht das Programm nun den NEXT-Befehl, so wird die Schleifenvariable um den ausdruck3 erhöht und dann mit dem ausdruck2 verglichen. Ist der Wert der Schleifenvariablen kleiner oder gleich dem ausdruck2, so wird die eingeschlossene Gruppe ein weiteres Mal ausgeführt, beginnend mit dem Befehl hinter FOR. In der ersten Form des Befehls ist die Erhöhung (STEP) gleich 1. In der zweiten Form wird die Erhöhung durch den ausdruck3 vorgegeben. Ist der Wert der Schleifenvariablen größer geworden als ausdruck2, so wird das Programm mit dem Befehl hinter NEXT fortgesetzt. Da der Vergleich am Ende der FOR/NEXT-Schleife durchgeführt wird, werden die eingeschlossenen Befehle mindestens einmal ausgeführt.

Der Wert von ausdruck1, ausdruck2 und ausdruck3 muß zwischen -9.999999999 E 99 und 9.999999999 E 99 liegen. Wird ausdruck3 gleich Ø gesetzt, so kann das Programm nicht fortgesetzt werden.

Die Schleifenvariable kann in der umschlossenen Gruppe von Befehlen benutzt werden. Mit ihr kann z. B. ein Feld-Index "hochgezählt" werden. Man sollte aber sehr vorsichtig sein, wenn der Wert der Schleifenvariablen verändert wird.

Programme dürfen niemals so geschrieben werden, daß von einem Befehl außerhalb einer FOR/NEXT-Schleife in eine solche hineingesprungen wird. Ein Programm sollte auch nicht aus einer FOR/NEXT-Schleife herausspringen. Sie sollten eine FOR/NEXT-Schleife immer über den NEXT-Befehl verlassen. Der Schleifenvariablenwert muß dazu nur größer als ausdruck2 gesetzt werden.

Eine Gruppe von Befehlen, die von einer FOR/NEXT-Schleife eingefaßt werden, kann eine weitere FOR/NEXT-Schleife enthalten. Diese muß dann eine andere Schleifenvariable haben und vollständig in der ersten Schleife liegen. Wenn also in einer FOR/NEXT-Schleife ein FOR-Befehl auftaucht, so muß auch der dazugehörigen NEXT-Befehl in der Schleife liegen. Sie können bis zu fünf FOR/NEXT-Schleifen ineinanderlegen.

### Beispiele:

10: FOR I=1 TO 5

Diese Schleife gibt die Zahlen 1, 2, 3, 4 und 5

20: PRINT I

aus.

30: NEXT I

zu Aufgabe 10: Ausgabe 1, 2, 3, 4, 5

40: FOR N=10 TO 0 STEP -1

Diese Schleife zählt rückwärts von 10 nach 0

50: PRINT N

und gibt die Schleifenvariable aus.

60: NEXT N

zu Aufgabe 40: Ausgabe 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0

70: FOR N=1 TO 10

Diese Schleife berechnet die Fakultäten der

80: X=1

Zahlen von 1 bis 10 und gibt das Ergebnis auf

90: FOR F=1 TO N

der Anzeige wieder.

100: X=X \* F

zu Aufgabe 90: Ausgabe 3,6,24,120,720,5040,40320,362880,3628800,36288000

110: NEXT F

zu Aufgabe 100: Ausgabe 3,6,24,120,720,5040,40320,362880,3628800,36288000

120: PRINT X

zu Aufgabe 120: Ausgabe 3,6,24,120,720,5040,40320,362880,3628800,36288000

130: NEXT N

zu Aufgabe 130: Ausgabe 3,6,24,120,720,5040,40320,362880,3628800,36288000

**Achtung:** Programme sollten nicht aus einer FOR/NEXT-Schleife herausspringen. Es könnte hierbei passieren, daß Sie eine Fehlermeldung (ERROR-5) im Programmablauf (bei Programmen, die mehrere FOR/NEXT-Schleifen enthalten) angezeigt bekommen.

Ein Beispiel für einen solchen Fehler ist das Programm "FOR-NEXT-Test". Es besteht aus einer FOR-Schleife, die eine weitere FOR-Schleife enthält. Diese innere Schleife hat die Variable I als Schleifenvariable. Wenn man nun die innere Schleife mit dem Befehl EXIT I verlässt, wird dies als Fehler erkannt.

Um diesen Fehler zu verhindern, kann man die innere Schleife mit dem Befehl EXIT N verlassen. Dieser Befehl verlässt die innere Schleife und kehrt in die äußere Schleife zurück. Der Befehl EXIT I verlässt die innere Schleife und kehrt in den Programmablauf zurück.

Ein weiteres Problem bei der Verwendung von Schleifen ist die Tatsache, daß es sich um geschachtelte Schleifen handelt. Wenn man eine innere Schleife mit dem Befehl EXIT I verlässt, wird dies als Fehler erkannt. Das gleiche gilt für die äußere Schleife.

## GOSUB

### 1. GOSUB ausdruck

Abkürzung: GOS., GOSU.

Vergleiche: GOTO, ON...GOSUB, ON...GOTO, RETURN

#### Wirkung:

Mit dem GOSUB-Befehl springt das Programm in eine BASIC-Unterroutine.

#### Anwendung:

Wenn eine Gruppe von Befehlen in einem Programm an verschiedenen Stellen ausgeführt werden soll oder diese Befehlsgruppe in verschiedenen Programmen eingesetzt werden soll, so empfiehlt sich die Verwendung der Befehle GOSUB und RETURN.

Die Befehlsgruppe wird so untergebracht, daß das Programm normalerweise nicht in diese Befehle hineinlaufen kann. Eine Möglichkeit wäre, diese Gruppe hinter den END-Befehl zu stellen. An den Stellen des Hauptprogramms, an der diese Gruppe nun ausgeführt werden soll, fügen Sie den GOSUB-Befehl mit der Zeilennummer der Startzeile der Unterroutine als ausdruck ein. Die letzte Zeile der Unterroutine muß einen RETURN-Befehl enthalten. Wird der GOSUB-Befehl nun ausgeführt, so wird das Programm in der Unterroutine fortgesetzt. Erreicht der Computer den RETURN-Befehl, so springt der Rechner in den Befehl nach dem GOSUB im Hauptprogramm zurück.

In einer Unterroutine kann ein GOSUB vorkommen. Maximal können 10 Unterroutinen ineinandergelegt werden.

Der ausdruck im GOSUB-Befehl darf kein Komma enthalten, z.B. A(2,5) kann nicht benutzt werden. Um von einem bestimmten Punkt mehrere Unterroutinen anspringen zu können, wird der Befehl ON...GOSUB benutzt. Der ausdruck kann nur eine erlaubte Zeilennummer sein, d. h. der Wert muß zwischen 1 und 65279 inklusive liegen oder Sie bekommen die Fehlermeldung ERROR 4.

#### Beispiel:

10: GOSUB 100

Das Programm springt von Zeile 10 in die Unterroutine in Zeile 100, gibt das Wort HALLO aus und springt dann in die Zeile 20 des Hauptprogramms zurück.

20: END

100: PRINT "HALLO"

110: RETURN

## GOTO

### 1. GOTO ausdruck

Abkürzung: G., GO., GOT.

Vergleiche: GOSUB, ON ... GOSUB, ON ... GOTO

#### Wirkung:

Mit dem GOTO-Befehl springt der Rechner an eine bestimmte Stelle seines Programms.

#### Anwendung:

Der Befehl veranlaßt den Computer, von einer bestimmten Stelle in einem Programm an eine andere, durch den ausdruck bestimmte Stelle des Programms zu springen. Anders als im GOSUB-Befehl „erinnert“ sich der Rechner nicht, von wo dieser Sprung ausgeführt wurde.

Der ausdruck des Befehls darf kein Komma, wie z.B. in A(1,2) enthalten. Soll von einem Punkt die Möglichkeit gegeben werden, an eine von mehreren Stellen zu springen, so muß der ON ... GOTO-Befehl benutzt werden. Der ausdruck muß eine Zeilennummer sein. Erlaubt sind die Werte von 1 bis 65279. Bei einem anderen Wert bekommen Sie die Fehlermeldung ERROR 4.

Gute Programme laufen von Anfang bis Ende ohne Sprünge, abgesehen von Unter-routinen-sprungen. Der prinzipielle Gebrauch des GOTO-Befehls ist deshalb im IF ... THEN-Befehl zu finden.

#### Beispiel:

```
10; INPUT A$  
20; IF A$="J" THEN GOTO 50  
30; PRINT "NEIN"  
40; GOTO 60  
50; PRINT "JA"  
60; END
```

**GRAD****1. GRAD**

Abkürzung: GR., GRA.

Vergleiche: DEGREE, RADIAN

**Wirkung:**

Mit dem GRAD-Befehl werden die Winkelwerte auf Neu-Grad umgeschaltet.

**Anwendung:**

Der Computer hat drei Formen der Winkeldarstellung:

1. Alt-Grad
2. Radial-Wert
3. Neu-Grad

Die Argumente der Funktionen SINus, COSinus und TANgents können in dieser Form eingegeben bzw. die Ergebnisse der Umkehrfunktionen ArcusSiNus, ArcusCoSinus und ArcusTaNgens in diesen Formen ausgegeben werden.

**Beispiel:**

10: GRAD                    X hat jetzt den Wert 100, den Gradienten von Arcussinus  
20: X=ASN 1                1.  
30: PRINT X

**IF . . . THEN**

1. IF bedingung THEN befehl
2. IF bedingung befehl

Abkürzung: keine für IF ; T., TH., THE.

Während der normalen  
Ausführung wird der Befehl ausgeführt

**Wirkung:**

Das IF . . . THEN-Befehlspaar wird benutzt, um einen bestimmten Befehl nur dann auszuführen, wenn eine Bedingung erfüllt ist.

**Anwendung:**

Im normalen Ablauf eines BASIC-Programms werden die Befehle in der Reihenfolge ihres Auftretens abgearbeitet. Das IF . . . THEN-Befehlspaar erlaubt es, einen Befehl nur dann auszuführen, wenn eine bestimmte Bedingung erfüllt ist. Ist der Bedingungsteil des IF-Befehls wahr, so wird der Befehl ausgeführt, ist die Bedingung falsch, wird er übersprungen.

Der Bedingungsteil des IF-Befehls kann jeder vergleichende Ausdrück sein. Es ist auch möglich, einen numerischen Ausdrück als Bedingung zu setzen, obgleich der Sinn des Befehls dadurch wenig klar wird. Jeder Ausdruck, der kleiner oder gleich Null ist, setzt die Bedingung falsch. Jeder positive Wert setzt die Bedingung wahr.

Der Befehl, der auf den THEN-Teil des Befehls folgt, kann jeder BASIC-Befehl sein, auch ein weiterer IF . . . THEN-Befehl. Handelt es sich um eine Zuweisung, so muß der LET-Befehl mit dem Wort LET eingeleitet werden. Ist der auf THEN folgende Befehl kein END, GOTO oder ON . . . GOTO, so wird der dann folgende Befehl ausgeführt, gleich, ob die Bedingung wahr oder falsch ist.

Die beiden Formen des Befehls sind in der Ausführung gleich, aber die erste Form ist wesentlich klarer.

**Beispiel:**

```

10: INPUT "WEITER?"; A$          10: INPUT "WEITER?"; A$
20: IF A$="JA" THEN GOTO 10      20: IF A$="JA" THEN GOTO 10
30: IF A$="NEIN" THEN GOTO 60    30: IF A$="NEIN" THEN GOTO 60
40: PRINT "JA ODER NEIN, BITTE" 40: PRINT "JA ODER NEIN, BITTE"
50: GOTO 10                      50: GOTO 10
60: END                          60: END

```

Dieses Programm wiederholt die Frage WEITER?, solange JA eingegeben wird. Es stoppt, wenn NEIN eingegeben wird. Andere Antworten werden nicht beachtet.

## INPUT

### 1. INPUT eingabe liste

eingabe liste    ist eingabe gruppe

oder

eingabe gruppe, eingabe liste

eingabe gruppe    ist var list

oder

prompt. var liste

oder

prompt; var liste

var liste    ist variable

oder

variable, var liste

prompt    ist jede String-Konstante

Abkürzung: I., IN., INP., INPU.

Vergleiche: INPUT #, READ, PRINT

#### Wirkung:

Der INPUT-Befehl ermöglicht es, einen oder mehrere Werte über die Tastatur einzugeben.

#### Anwendung:

Wollen Sie in einem Programm die Grundwerte für jeden Programmlauf ändern, so empfiehlt sich der INPUT-Befehl. Die Basiswerte können dann mit der Tastatur eingegeben werden.

In der einfachsten Form fordert der Befehl Sie mit einem Fragezeichen in der linken Ecke der Anzeige auf, die nötige Eingabe zu machen. Sie geben dann den gewünschten Wert, gefolgt von **ENTER** ein. Dieser Wert wird dann der ersten Variablen der Variablenliste des INPUT-Befehls zugewiesen. Sind in dem Befehl mehrere Variable aufgelistet, so wird der Prozeß so oft wiederholt, bis alle Variablen belegt sind.

Sie können die Eingabe-Aufforderung (= Prompt), also das Fragezeichen, durch einen Prompt-String, das ist eine Bemerkung, die die nötige Eingabe erklären sollte, eigener Wahl ersetzen. Der Ablauf des Zuweisungsprozesses ändert sich nur insofern, als das Fragezeichen durch diesen String ersetzt wird.

Sollte der Befehl einen Prompt-String und die Variablenliste mehr als eine Variable enthalten, so wird der String nur bei der ersten Aufforderung gezeigt. Die folgenden werden wieder mit dem Fragezeichen gekennzeichnet. Beinhaltet der Befehl einen zweiten Prompt-String, so wird dieser vor der Zuweisung der direkt folgenden Variablen angezeigt.

Wenn Sie bei einem INPUT keinen Wert eingeben und direkt die **ENTER** -Taste drücken, so behält die entsprechende Variable ihren alten Wert.

### **Beispiele:**

## 10: INPUT A

Löscht das Anzeigenfeld und setzt ein Fragezeichen an den Anfang der 1. Zeile

20: INPUT "A="; A

Gibt "A=" aus und wartet auf eine Eingabe.

30: INPUT "A=", A

Gibt "A=" aus und wartet auf eine Eingabe.  
Gibt "A=" aus, wartet auf eine Eingabe und  
zeigt diese dann am Anfang der 1. Zeile an.  
Der Prompt-String wird also gelöscht.

40: INPUT "X=?": X: "Y=?": Y

Gibt erst das Prompt "X=?" aus, wartet auf eine Eingabe, löscht nach dem **ENTER** die Anzeige und gibt dann den Prompt "Y=?" aus.

**INPUT #**

1. INPUT # var liste
2. INPUT # "dateiname"; var liste

var liste      ist variable      oder  
variable, var liste

Abkürzung: I. #, IN. #, INP. #, INPU. #

Vergleiche: INPUT, PRINT #, READ

**Wirkung:**

Mit dem INPUT #-Befehl können Daten von einer Cassette geladen werden.

**Anwendung:**

Folgende Variable können im INPUT #-Befehl eingesetzt werden:

1. gesetzte Variable – A, B, C, A(7), D\*, A(20)\* usw.
2. einfache Variable – AA, B3, CP\$ usw.
3. Feld-Variable – S(\*), HP(\*), K\$(\*) usw.

**1) Übergabe von Daten auf gesetzte Variable**

Um Daten von Cassette auf gesetzte Variable zuzuweisen, müssen die Namen der Variablen im Befehl angegeben werden.

INPUT# "DATA1"; A, B, X, Y

Der Befehl weist die Daten der Datei DATA1 den Variablen A, B, X und Y in der gegebenen Reihenfolge zu.

Um alle verfügbaren gesetzten Variablen und, falls definiert, erweiterte Variable (A(27) und weiter) mit Daten von Cassette zu belegen, muß der ersten Variablen ein Stern (\*) angehängt werden.

INPUT# "D2"; D\*

Mit diesem Befehl werden die Daten der Cassette den Variablen D bis Z und A(27) und höher zugewiesen.

INPUT# A(10)\* (ohne DIMensionierung)

Dieser Befehl überträgt die Daten der ersten gefundenen Datei von Cassette auf die Variablen A(10) und höher (also J bis Z und A(27) und höher).

**Achtung:**

- a) Wurde bereits mit einem DIM-Befehl ein Feld namens "A" definiert, kann keine erweiterte Variable der Form A( ) definiert werden.

- b) Die Übertragung der Daten auf die gesetzten und erweiterten Variablen (A(27) und höher) wird so lange fortgesetzt, wie Daten gefunden werden, oder bis der Speicher des Rechners voll ist.

### 2) Datenübertragung auf einfache Variable

Werden im INPUT#-Befehl einfache Variablennamen eingegeben, so werden die Daten der Cassette auf diese Variablen übertragen.

**INPUT# "DM!"; AB, Y1, XY\$**

Dieser Befehl überträgt die Daten von Cassette auf die Variablen AB, Y1 und XY\$.

#### Achtung:

- Numerische Daten müssen auf numerische Variable, String-Daten auf String-Variable übertragen werden. Andere Zuweisungen sind nicht möglich.
- Im Programm-Datenspeicherbereich muß Raum für die einfachen Variablen frei gehalten werden, bevor der INPUT #-Befehl ausgeführt werden kann, sonst erhalten Sie eine Fehlermeldung. Die "Platzreservierung" kann mit einer Zuweisung ausgeführt werden.

**AA=0 [ENTER]**

**BI\$="A" [ENTER]**

**INPUT# AA, BI\$ [ENTER]**

Für die Zuweisung können einfache numerische oder String-Werte benutzt werden.

### 3) Datenübertragung auf Feldvariable

Um Daten von Cassette auf die Variablen eines Feldes zu übertragen, muß der Name des Feldes in der Form feldname (\*) im Befehl angegeben werden.

**50: DIM B(5)**

**60: INPUT # "DS4"; B(\*)**

Dieser Befehl überträgt die Daten der Datei DS4 auf die Variablen B(0) bis B(5) des Feldes B.

#### Achtung:

- Numerische Daten müssen auf numerische Feldvariablen gleicher Länge, String-Daten müssen auf String-Feldvariable gleicher Länge übertragen werden. Wird dies nicht beachtet, erhalten Sie eine Fehlermeldung.
- Sie müssen Raum im Programm-Datenspeicherbereich bereitstellen, bevor der INPUT#-Befehl ausgeführt wird oder Sie erhalten eine Fehlermeldung. Benutzen Sie dazu den DIM-Befehl.

### WICHTIG

Stimmt die Anzahl der Variablen im INPUT#-Befehl nicht mit der Zahl der Daten auf der Cassette überein, geschieht folgendes:

- \* Ist die Zahl der Daten in der Datei auf Cassette größer als die Zahl der Variablen des Befehls, so wird jeder Variablen ein Wert zugewiesen. Die überzähligen Daten werden ignoriert.
- \* Ist die Zahl der Daten in der Datei auf Cassette kleiner als die Zahl der Variablen des Befehls, so werden alle Daten der Datei den Variablen zugewiesen und die überzähligen Variablen behalten ihren alten Wert.  
Der Rechner wartet aber auf weitere Daten für diese Variablen. Unterbrechen Sie diesen Status mit der **BRK**-Taste.
- \* Wird der INPUT#-Befehl ohne Variable eingegeben, so erhalten Sie bei dem Versuch der Ausführung die Fehlermeldung ERROR 1.

**LET**

SINUSOID

**1. LET variable = ausdruck****2. variable = ausdruck****Abkürzung:** LE,

Von der Abkürzung LE kann man leicht merken, daß es sich um einen Zuweisungsbefehl handelt.

**Wirkung:**

Der LET-Befehl weist Variablen Werte zu.

Von den beiden Varianten kann man leicht merken, daß es sich um eine Zuweisung handelt.

**Anwendung:**

Mit dem LET-Befehl weisen Sie den angegebenen Variablen den Wert des Ausdrucks zu. Sie können einer numerischen Variablen nur numerische Werte und einer String-Variablen nur String-Werte zuweisen. Um Werte aus der einen in die andere Form zu übertragen, muß eine der beiden Funktionen STR\$ oder VAL eingesetzt werden.

Sie können das LET in einer Zuweisung weglassen, es sei denn, die Zuweisung folgt auf das THEN eines IF . . . THEN-Befehles. Dies ist der einzige Fall, wo das Wort LET im Befehl vorkommen muß.

**Beispiele:**

10: I=10

Weist I den Wert 10 zu.

20: A=5\*I

Weist A den Wert 50 zu.

30: X\$=STR\$(A)

Weist X\$ den Wert "50" zu.

40: IF I&gt;=10 THEN LET Y\$=X\$+"00"

Weist Y\$ den Wert "50.00" zu.

## Für Drucker CE-126P

### LPRINT

1. LPRINT druckausdr
2. LPRINT druckausdr, druckausdr
3. LPRINT druckliste; druckausdr; . . . ; druckausdr

Abkürzung: LP., LPR., LPR1., LPRIN.

Vergleiche: PRINT, USING

#### Wirkung:

Mit dem LPRINT-Befehl werden Informationen auf der Option CE-126P ausgegeben.

#### Anwendung:

Wenn als Folge des OPEN-Kommandos der serielle E/A-Anschluß geöffnet ist, erreichen Sie mit dem LPRINT-Kommando eine Ausgabe auf dem dort angegeschlossenen Drucker. Um die Ausgabe des Programms auf dem Drucker durchzuführen, geben Sie ein CLOSE-Kommando ein.

- \* In der ersten Form des Befehls werden numerische Werte rechtsbündig und alphabetische Zeichen vom linken Rand des Papiers ausgedruckt. Wenn eine Zeile mehr als 24 Zeichen enthält, wird automatisch ein Zeilenvorschub-Befehl ausgeführt.
- \* In der Form 2 werden die 24 Zeichen einer Zeile in zwei Gruppen zu je 12 Zeichen aufgeteilt. Die Daten werden symmetrisch vom Komma ausgedruckt.

Werte, die mehr als 12 Zeichen lang sind, werden auf 12 Zeichen gekürzt. Numerische werden dabei in Dezimalbruch und Zehnerpotenz ausgegeben. Strings werden nur bis zum 12. Zeichen ausgedruckt.

From 3 drückt die Werte links beginnend aus. Werden mehr als 24 Zeichen eingegeben, so wird der Druck in der nächsten Zeile fortgesetzt. Maximal können 96 Zeichen ausgedruckt werden. Liegt die 96.

- \* Der Drucker CE-126P hat Priorität, wenn sowohl der CE-126P und der CE-140P angeschlossen sind.

**Beispiele:**

```

10: A=10: B=20: X$="ABCD": Y$="XYZ"
20: LPRINT A
30: LPRINT X$
40: LPRINT A, B, X$, Y$
50: LPRINT X$; A; B
60: LPRINT
70: LPRINT A * B;
80: LPRINT Y$
```

Ergebnis:  
10  
20  
ABCD  
40 10 20 ABCD XYZ  
50 ABCD; 10; 20  
60  
70 200  
80 XYZ

Die Ergebnisse der obigen Beispiele sind in Abbildung 10-1 dargestellt.

Die Anweisungen 10 bis 80 sind im Bildschirm abgespeichert. Die Anweisung 10 definiert die Variablen A und B mit den Werten 10 und 20. Die Variable X\$ erhält den Wert "ABCD". Die Variable Y\$ erhält den Wert "XYZ". Die Anweisung 20 ruft die Variable A auf. Die Anweisung 30 ruft die Variable X\$ auf. Die Anweisung 40 ruft die Variablen A, B, X\$ und Y\$ auf. Die Anweisung 50 ruft die Variablen X\$, A und B auf. Die Anweisung 60 ruft keine Variablen auf. Die Anweisung 70 ruft die Variable A\*B auf. Die Anweisung 80 ruft die Variable Y\$ auf.

Die Ergebnisse der obigen Beispiele sind in Abbildung 10-1 dargestellt. Die Anweisungen 10 bis 80 sind im Bildschirm abgespeichert. Die Anweisung 10 definiert die Variablen A und B mit den Werten 10 und 20. Die Variable X\$ erhält den Wert "ABCD". Die Variable Y\$ erhält den Wert "XYZ". Die Anweisung 20 ruft die Variable A auf. Die Anweisung 30 ruft die Variable X\$ auf. Die Anweisung 40 ruft die Variablen A, B, X\$ und Y\$ auf. Die Anweisung 50 ruft die Variablen X\$, A und B auf. Die Anweisung 60 ruft keine Variablen auf. Die Anweisung 70 ruft die Variable A\*B auf. Die Anweisung 80 ruft die Variable Y\$ auf.

Die Ergebnisse der obigen Beispiele sind in Abbildung 10-1 dargestellt. Die Anweisungen 10 bis 80 sind im Bildschirm abgespeichert. Die Anweisung 10 definiert die Variablen A und B mit den Werten 10 und 20. Die Variable X\$ erhält den Wert "ABCD". Die Variable Y\$ erhält den Wert "XYZ". Die Anweisung 20 ruft die Variable A auf. Die Anweisung 30 ruft die Variable X\$ auf. Die Anweisung 40 ruft die Variablen A, B, X\$ und Y\$ auf. Die Anweisung 50 ruft die Variablen X\$, A und B auf. Die Anweisung 60 ruft keine Variablen auf. Die Anweisung 70 ruft die Variable A\*B auf. Die Anweisung 80 ruft die Variable Y\$ auf.

Die Ergebnisse der obigen Beispiele sind in Abbildung 10-1 dargestellt. Die Anweisungen 10 bis 80 sind im Bildschirm abgespeichert. Die Anweisung 10 definiert die Variablen A und B mit den Werten 10 und 20. Die Variable X\$ erhält den Wert "ABCD". Die Variable Y\$ erhält den Wert "XYZ". Die Anweisung 20 ruft die Variable A auf. Die Anweisung 30 ruft die Variable X\$ auf. Die Anweisung 40 ruft die Variablen A, B, X\$ und Y\$ auf. Die Anweisung 50 ruft die Variablen X\$, A und B auf. Die Anweisung 60 ruft keine Variablen auf. Die Anweisung 70 ruft die Variable A\*B auf. Die Anweisung 80 ruft die Variable Y\$ auf.

Die Ergebnisse der obigen Beispiele sind in Abbildung 10-1 dargestellt. Die Anweisungen 10 bis 80 sind im Bildschirm abgespeichert. Die Anweisung 10 definiert die Variablen A und B mit den Werten 10 und 20. Die Variable X\$ erhält den Wert "ABCD". Die Variable Y\$ erhält den Wert "XYZ". Die Anweisung 20 ruft die Variable A auf. Die Anweisung 30 ruft die Variable X\$ auf. Die Anweisung 40 ruft die Variablen A, B, X\$ und Y\$ auf. Die Anweisung 50 ruft die Variablen X\$, A und B auf. Die Anweisung 60 ruft keine Variablen auf. Die Anweisung 70 ruft die Variable A\*B auf. Die Anweisung 80 ruft die Variable Y\$ auf.

## Für Drucker CE-140P

### LPRINT

1. LPRINT druckausdr
2. LPRINT druckausdr, druckausdr, . . . , druckausdr
3. LPRINT druckausdr; druckausdr; . . . ; druckausdr
4. LPRINT . . . druckausdr;

Bei dieser Form wird an das Ende von 1 und 3 oben ein ";" gesetzt.

5. LPRINT

Abkürzung: LP., LPR., LPRI., LPRIN.

Vergleiche: PRINT, CONSOLE, USING

### Wirkung:

Mit dem Befehl LPRINT werden Informationen auf den Drucker ausgegeben.

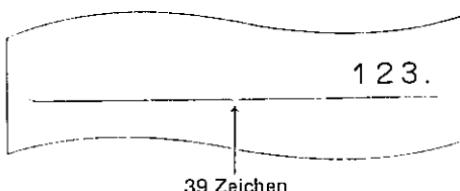
### Anwendung:

Wenn mit dem OPEN-Kommando der serielle E/A-Anschluß geöffnet wurde, gibt das LPRINT-Kommando das Programm am seriellen E/A-Schnittstellen-Anschluß aus. (Siehe Seite 199.) Zur Rückkehr des Programm-Druck-Kommandos zum CE-140P das CLOSE-Kommando ausführen. (Siehe Seite 191.)

In Form 1 werden numerische Werte rechtsbündig und alphabetische Zeichen beginnend vom linken Papierrand innerhalb des mit dem CONSOLE-Kommando spezifizierten Bereiches (normalerweise 39 Zeichen) gedruckt. (Für Informationen zum CONSOLE-Kommando siehe Seite 192.)

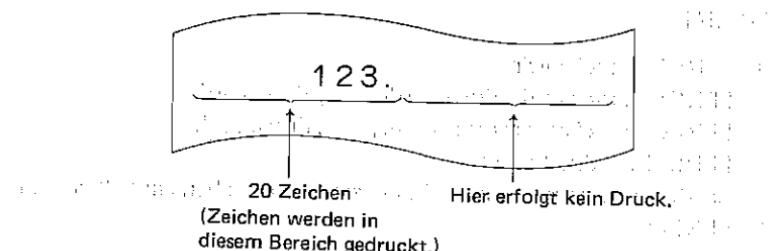
Beispiel: LPRINT 123 **[ENTER]** wird ausgeführt mit der Standard-Zeichengröße (Größe b)

1. Wenn 39 Zeichen pro Druckzeile mit dem CONSOLE-Kommando spezifiziert wurden



(Zeichen werden in diesem Bereich gedruckt.)

**2. Wenn 20 Zeichen pro Druckzeile mit dem CONSOLE-Kommando spezifiziert wurden**



In Form (2) ist die Anzahl Druckzeichen in Gruppen von 12 Zeichen unterteilt. Die spezifizierten Werte werden der Reihe nach ausgedruckt. Numerische Werte mit bis zu 12 Ziffern werden rechts und Zeichenstrings links auf dem Drucker ausgegeben. Wenn der auszudruckende Wert 12 Zeichen überschreitet, werden numerische Werte auf 12 Zeichen gerundet, während von Zeichenstrings die ersten 12 Zeichen (von links) gedruckt werden.

Die Maximale Anzahl Werte (Elemente, die spezifiziert und durch Kommata getrennt werden können) beträgt für Form (2) 8.

Werden mehr Elemente spezifiziert, tritt die Fehlermeldung ERROR 1 auf.

In Form (3) werden die Werte beginnend vom linken Papierrand ausgedruckt. Wenn der auszudruckende Werte 24 Zeichen überschreitet, wird automatisch ein Zeilenvorschub ausgeführt. Bis zu maximal 96 Zeichen können gedruckt werden. Ein Fehler tritt auf, wenn das 96.

In Form (4) wird durch das Semikolon der Startpunkt des nächsten LPRINT-Befehls bestimmt. Dieser befindet sich direkt hinter dem zuletzt ausgegebenen Wert.

Hinweis: Wenn sich nur ein zu druckender Wert in einer durch einen LPRINT-Befehl zu druckenden Gleichung befindet, der auf einen in Form (4) geschriebenen Befehl folgt, diesen Wert mit ; " ; " nach der Gleichung spezifizieren.

Beispiel:

```
10 A=50 : B=80
20 LPRINT "A * B=" ;
30 LPRINT A * B ; " ; "
```

Anmerkung: " ; " verwenden, um Ergebnisse zu spezifizieren, die mit Zeile 30 nach dem Drucken in Zeile 20 gedruckt werden sollen.

Die Form (5) bewirkt kein Drucken, sondern nur einen Zeilenvorschub.

## NEXT

### 1. NEXT numerische variable

Abkürzung: N., NE., NEX.

Vergleiche: FOR

#### Wirkung:

Der NEXT-Befehl kennzeichnet das Ende einer Gruppe von Befehlen, die in einer FOR/NEXT-Schleife wiederholt werden sollen.

#### Anwendung:

Der NEXT-Befehl tritt immer in Verbindung mit einem FOR-Befehl auf und wird deshalb dort ausführlich erklärt. Die Variable des NEXT muß mit der des dazugehörigen FOR übereinstimmen.

#### Beispiel:

10: FOR I=1 TO 10

20: PRINT I

30: NEXT I

**ON ... GOSUB**

1. ON ausdruck GOSUB ausdr liste

ausdr liste      ist ausdruck

oder

ausdruck, ausdr liste

Abkürzung: O.; GOS.; GOSU.

Vergleiche: GOSUB, GOTO, ON ... GOTO

**Wirkung:**

Der ON ... GOSUB-Befehl wird benutzt, um eine der gegebenen Unterroutine abhängig von einem Wert auszuführen.

**Anwendung:**

Bei der Ausführung des ON ... GOSUB-Befehls wird der ausdruck zwischen ON und GOSUB auf einen ganzzahligen Wert reduziert. Ist der Wert dieses Ausdrucks 1, so wird die erste der durch ausdr liste angegebenen Unterroutine nach dem GOSUB ausgeführt. Ist der Wert 2, so wird die zweite ausgeführt usw. Nach dem RETURN aus der Unterroutine wird der Befehl der direkt auf den ON ... GOSUB-Befehl folgt, ausgeführt.

Wenn der Wert des Ausdrucks kleiner 1 oder größer der Zahl der gegebenen Unterroutine ist, wird das Programm mit dem auf den ON ... GOSUB-Befehl folgenden Befehl fortgesetzt.

Achtung: In den Ausdrücken nach dem GOSUB dürfen keine Kommata vorkommen. Der Computer kann nicht Kommata in Ausdrücken von Kommata zwischen Ausdrücken unterscheiden.

**Beispiel:**

10: INPUT A	Wird in Zeile 10 "1" eingegeben, so gibt der Rechner "ERSTE" aus. Auf "2" gibt er "ZWEITE" und auf "3" "DRITTE" aus. Auf andere Eingaben wird nichts ausgegeben.
20: ON A GOSUB 100, 200, 300	
30: END	
100: PRINT "ERSTE"	
110: RETURN	
200: PRINT "ZWEITE"	
210: RETURN	
300: PRINT "DRITTE"	
310: RETURN	

ON . . . GOTO

1. ON ausdruck GOTO ausdr liste

## **ausdr liste      ausdruck**

oder

**ausdruck, ausdr liste**

Abkürzung: O.; G.; GO.; GOT.

Vergleiche: GOSUB, GOTO, ON . . . GOSUB

### **Wirkung:**

Mit dem Befehl ON . . . GOTO wird es möglich, das Programm abhängig von einem Wert an einer bestimmten Stelle fortzusetzen.

### Anwendung:

Bei der Ausführung des Befehls ON . . . GOTO wird der ausdruck zwischen ON und GOTO auf einen ganzzahligen Wert reduziert. Ist der Wert dieser Zahl 1, so wird das Programm an der ersten angegebenen Stelle fortgesetzt. Ist er 2, an der zweiten usw.

Ist der Wert des Ausdrucks kleiner 1 oder größer der Zahl der gegebenen Sprungziele, so wird der auf den ON ... GOTO-Befehl folgende Befehl ausgeführt.

Achtung: In den Ausdrücken, die auf das GOTO folgen, dürfen keine Kommata vorkommen. Der Computer ist nicht in der Lage, Kommata, die in Ausdrücken auftreten, von Kommata, die zwischen Ausdrücken stehen, zu unterscheiden.

### **Beispiel:**

```
10: INPUT A
20: ON A GOTO 100, 200, 300
30: GOTO 900
100: PRINT "ERSTE"
110: GOTO 900
200: PRINT "ZWEITE"
210: GOTO 900
300: PRINT "DRITTE"
900: END
```

Die Eingabe '1' lässt den Rechner 'ERSTE', '2' 'ZWEITE' und '3' 'DRITTE' ausgeben.

**PAUSE**

1. PAUSE druck ausdr
2. PAUSE druck ausdr, druck ausdr
3. PAUSE druck liste; druck liste; ... ; druck list

druck listedruck ausdr

oder

druck ausdr; druck listeausdruck

oder

USING format; ausdruckDas USING format wird unter USING beschrieben.

Abkürzung: PAU., PAUS.

Vergleiche: LPRINT, PRINT, USING, WAIT

**Wirkung:**

Mit dem PAUSE-Befehl wird eine Ausgabe auf dem Display zeitlich begrenzt wiedergegeben.

**Anwendung:**

Der PAUSE-Befehl wird benutzt, um kurze Texte, Ergebnisse usw. anzuzeigen. Die Ausführung des Befehls ist die gleiche wie die des PRINT-Befehls. Mit einer Ausnahme: Nach der Ausgabe des Textes oder des Wertes macht der Computer eine Pause von ca. 0,85 Sekunden, bevor der nächste Befehl ausgeführt wird. Ein WAIT-Intervall oder **ENTER** wird nicht gebraucht.

Die erste Form des PAUSE-Befehls gibt einen einzelnen Wert aus. Numerische Werte werden ganz rechts, String-Werte ganz links beginnend ausgegeben.

In Form 2 wird das Anzeigenfeld in Gruppen mit je 12 Spalten unterteilt. Die Werte werden vom ersten spezifizierten Wert an in aufsteigender Reihenfolge angegeben. Auch hier werden innerhalb des Blocks numerische Werte ganz links und String-Werte ganz rechts beginnend ausgegeben.

\* Die Anzahl der Werte in Form 2 darf höchstens 2 betragen.

\* Ist ein Wert mehr als 12 Spalten (Zeichen) lang, wird

- a) ein numerischer Wert als Dezimalbruch mit Exponent wiedergegeben,
- b) ein Stringwert auf seine ersten 12 Zeichen gekürzt.

Form 3 gibt die Werte in direkter Folge von der linken Seite des Bildschirms, beginnend wieder.

**Achtung:** Wenn der eingegebene Wert von Form 3 das Maximum von 96 Spalten überschreitet, wird der überschüssige Teil nicht angezeigt. Wenn der errechnete Wert das Maximum von 96 Spalten übersteigt, wird die Fehlermeldung 6 ausgegeben, sofern die 96. Spalte in einem numerischen Wert liegt.

**Beispiele:**

10: A=10 : B=20 : X\$="ABCDEF" : Y\$="XYZ"

**ANZEIGE**

20: PAUSE A		10.
30: PAUSE X\$	A B C D E F	
40: PAUSE X\$, B	A B C D E F	20.
50: PAUSE Y\$; X\$	X Y Z A B C D E F	
60: PAUSE A * B		200

**PRINT**

1. PRINT druck ausdr
2. PRINT durck ausdr, durck ausdr
3. PRINT druck liste
4. PRINT = LPRINT
5. PRINT = PRINT

druck listedruck ausdr

oder

druck ausdr; druck listedruck ausdrausdruck

oder

USING format; ausdruckDas USING format wird bei USING beschrieben.

Abkürzung: P., PR., PRI., PRIN.

Vergleiche: LPRINT, USING, WAIT

**Wirkung:**

Mit dem PRINT-Befehl können Ausgaben auf der Anzeige des Computers oder einen Drucker gemacht werden.

**Anwendung:**

Mit dem PRINT-Befehl werden Text und Ergebnisse ausgegeben. Die erste Form des Befehls gibt einen einzelnen Wert aus. Numerische Werte werden ganz rechts, String-Werte ganz links beginnend wiedergegeben.

Die Form 2 des Befehls teilt die Anzeige in Gruppen zu je 12 Spalten. Die Werte werden vom ersten spezifizierten Wert aus in aufsteigender Reihenfolge angezeigt. Die Startposition der Werte in den Feldern ist, unabhängig von irgendwelchen Vorgaben, für numerische Werte ganz rechts, für String-Werte ganz links.

- \* Die Anzahl der Werte in Form 2 darf höchstens 2 betragen.
- \* Ist ein Wert länger als 12 Spalten (Zeichen), so wird
  - a) ein numerischer Wert als Dezimalbruch mit Exponent wiedergegeben,
  - b) ein String-Wert auf die ersten 12 Zeichen gekürzt ausgegeben.

In Form 3 des Befehls werden die Werte direkt hintereinander links am Bildschirm wiedergegeben.

**Beispiele:**

10: A=123 : B=456 : X\$="ABCDEF" : Y\$="VWXYZ"

		ANZEIGE
20: PRINT X\$, B		A B C D E F                  4 5 6 .
30: PRINT A; B		1 2 3 . 4 5 6 .
40: PRINT X\$; A		A B C D E F 1 2 3 .
50: PRINT Y\$; B		V W X Y Z 4 5 6 .

**PRINT #**

1. PRINT # var liste
2. PRINT # "dateiname"; var liste

var liste      variable

oder variable

variable, var liste

Abkürzung: P.#, PR.#, PRI.#, PRIN.#

Vergleiche: INPUT#, PRINT, READ

**Wirkung:**

Mit dem PRINT #-Befehl werden Werte auf Cassette abgespeichert.

**Anwendung:**

Folgende Variabtentypen können benutzt werden:

1. gesetzte Variable — A, B, X, A(26), C\*, A(10)\* usw.
2. einfache Variable — AA, B2, XY\$ usw.
3. Feld-Variable — B(\*), CD(\*), N\$(\*) usw.

### 1) Abspeichern von gesetzten Variablen

Um die Werte von gesetzten Variablen auf Cassette abzuspeichern, müssen die Namen der Variablen im PRINT #-Befehl angegeben werden.

PRINT# "DATEI1"; A, B, X, Y

Dieser Befehl speichert die Inhalte der Variablen A, B, X und Y in der Datei "DATEI1" auf Cassette ab.

Wenn Sie den Inhalt aller Variablen ab einer bestimmten abspeichern wollen, so geben Sie die Startvariable gefolgt von einem Stern (\*) ein.

PRINT# "D2"; D\*

Dieser Befehl speichert die Inhalte der Variablen D bis Z (und A(26) und höher, wenn definiert) in der Datei mit dem Dateinamen "D2" ab.

PRINT# E, X\$, A(30)\*

Dieser Befehl speichert die Inhalte der gesetzten Variablen E und X\$ sowie die Inhalte der erweiterten Variablen A(30) und höher auf Cassette ab.

**Achtung:** A( ) kann nicht zur Definition nachfolgender Variablen benutzt werden, wenn A bereits durch einen DIM-Befehl definiert wurde. Ansonsten können die Variablen A(1) bis A(26) im PRINT#-Befehl genauso angegeben werden, wie die Variablen A bis Z bzw. A\$ bis Z\$.

## 2) Abspeicherung der Inhalte von einfachen Variablen

Die Inhalte von einfachen Variablen (Zwei-Zeichen-Variablen) werden abgespeichert, indem die Namen der Variablen im PRINT#-Befehl angegeben werden.

PRINT# "DM1"; AB, Y1, XY\$

Dieser Befehl speichert die Inhalte der einfachen Variablen AB, Y1 und XY\$ in der Datei mit dem Dateinamen "DM1" ab.

## 3) Abspeicherung der Inhalte von Feldvariablen

Die Inhalte aller Variablen eines Feldes können auf Cassette abgespeichert werden, indem der Name des Feldes im Befehl angegeben und die Variablen selbst mit einem Stern (\*) umschrieben werden.

PRINT# "DS2"; X(\*), Y(\*)

Dieser Befehl speichert die Inhalte der Elemente der Felder X (X(0), X(1), ...) und Y (Y(0), Y(1) ...) in der Datei mit dem Dateinamen "DS2" ab.

**Achtung:** Es ist nicht möglich, den Inhalt eines oder einiger Elemente eines Feldes gesondert abzuspeichern.

Während gesetzte und erweiterte Variable zulassen, daß nur ein spezifizierter Teil von ihnen auf Cassette gespeichert wird, ist dies mit einem durch den DIM-Befehl definierten Feld nicht möglich.

\* Wird der PRINT#-Befehl ohne Variable eingegeben, erhalten Sie die Fehlermeldung ERROR 1.

## WICHTIG

Die Lokation von erweiterten (A(26) und höher), einfachen und/oder Feldvariablen muß vor Ausführung des PRINT#-Befehls im Programm/Datenspeicherbereich bereitgestellt werden. Sonst ist die Ausführung des Befehls nicht möglich und Sie erhalten eine Fehlermeldung.

**RADIAN**

Umrechnung von Winkelwerten aus Grad in Radialform (Bogenmaß)

**1. RADIAN**

Umrechnung von Winkelwerten aus Radialform (Bogenmaß) in Grad

Abkürzung: RAD., RADI., RADIA.

Vergleiche: DEGREE, GRAD

**Wirkung:**

Der RADIAN-Befehl wechselt die Darstellung der Winkelwerte in die Radialform (Bogenmaß).

**Anwendung:**

Der Computer hat drei Möglichkeiten der Darstellung von Winkelwerten:

- Alt-Grad
- Radial-Wert
- Neu-Grad

Sie werden für die Argumente der Funktionen SINus, COSinus und TANgens sowie für die Ergebnisse der Umkehrfunktionen ArcusSiNus, ArcusCoSinus und ArcusTaNgens gebraucht.

Der RADIAN-Befehl wechselt die Darstellungsform auf Radialwert. Die Radialform gibt den Winkel als Bogenmaß A in Abhängigkeit vom Radius wieder. 360 Grad sind z. B.  $2 * \pi$ , da der Einheitskreis den Umfang 2 mal  $\pi$  mal Radius hat.

**Beispiel:**

10: RADIAN

20: X=ASN 1

30: PRINT X

X hat den Wert 1.570796327 (= $\pi/2$ ), Arcsin von 1.

## RANDOM

### 1. RANDOM

Abkürzung: RA., RAN., RAND., RANDO.

#### Wirkung:

Der RANDOM-Befehl setzt eine neue Startzahl für den Zufallszahlengenerator.

#### Anwendung:

Wenn mit der RND-Funktion Zufallszahlen erzeugt werden, so startet der Computer bei einer vorgegebenen Zahl. Der RANDOM-Befehl setzt die Startzahl auf einen neuen, zufällig gewählten Wert.

Die Startzahl für den Zufallszahlengenerator ist jedes Mal, wenn der Computer eingeschaltet wird, die gleiche. Auch die Sequenz der Zufallszahlen wiederholt sich damit, es sei denn, die Basis (oder Start-) Zahl wird mit dem RANDOM-Befehl gewechselt. Diese Eigenschaft ist sehr wichtig, ist es doch möglich, den korrekten Ablauf eines Programms zu ermitteln, da auch dann, wenn im Programm RND-Befehle auftreten, das Verhalten des Rechners immer gleich sein muß. Mit dem RANDOM-Befehl verbessern Sie also das Zufallsverhalten Ihres Programms, da die Basiszahl des Zufallsgenerators auch zufällig ermittelt wird.

#### Beispiel:

10: RANDOM  
20: X=RND 10  
30: GOTO 20

Bei Start in Zeile 20 wird die Standardbasis des Zufallszahlengenerators eingesetzt. Bei Start in Zeile 10 wird eine neue Basis ermittelt.

**READ****1. READ var liste**

var liste      variable  
                   oder

variable, var liste

**Abkürzung:** REA.

**Vergleiche:** DATA, RESTORE

**Wirkung:**

Der READ-Befehl wird gebraucht, um Daten aus den DATA-Zeilen herauszulesen und Variablen zuzuweisen.

**Anwendung:**

Wenn einem Feld Basiswerte zugewiesen werden sollen, so ist es sinnvoll, diese Daten in DATA-Zeilen unterzubringen und sie von dort mit einem READ-Befehl in einer FOR/NEXT-Schleife in das Feld zu übertragen. Wird der erste READ-Befehl ausgeführt, so wird der erste Wert aus den DATA-Zeilen zugewiesen, mit dem zweiten READ-Befehl der zweite Wert usw., unabhängig davon, wieviele DATA-Zeilen oder Werte in den DATA-Zeilen vorhanden sind.

Wenn nötig, können dieselben Daten ein zweites Mal gelesen werden. Hierzu wird der RESTORE-Befehl gebraucht.

**Beispiel:**

```

10: DIM B(10)
20: WAIT 32
30: FOR I=1 TO 10
40: READ B(I)
50: PRINT B(I) * 2
60: NEXT I
70: DATA 10,20,30,40,50,60
80: DATA 70,80,90,100
90: END

```

**Dimensionierung eines Feldes**

**Zuweisung der Werte aus den DATA-Zeilen in die Feldvariablen B(1) bis B(10); B(1) ist 10, B(2) ist 20 usw.**

## REM

### 1. REM Kommentar

#### Wirkung:

Mit dem REM-Befehl können Kommentare in ein Programm eingefügt werden.

#### Anwendung:

Es ist sinnvoll, in ein Programm erläuternde Kommentare einzufügen. Es kann sich hierbei um Titel, Autorennamen, Daten der letzten Änderung, Anwendungshinweise usw. handeln.

Solche Kommentare können mit einem REM-Befehl eingefügt werden. Der REM-Befehl hat keinen Einfluß auf den Programmablauf. Er kann überall im Programm eingefügt werden.

#### Beispiel:

10: REM DIESE ZEILE HAT KEINEN EFFEKT

**RESTORE**

1. RESTORE

2. RESTORE ausdruck

Abkürzung: RES., REST., RESTO., RESTOR.

Vergleiche: DATA, READ

**Wirkung:**

Der RESTORE-Befehl ermöglicht es, eine DATA-Zeile mehrmals zu lesen oder die Reihenfolge, in der die DATA-Zeilen gelesen werden sollen, zu verändern.

**Anwendung:**

Normalerweise werden mit dem READ-Befehl die Daten aus den DATA-Zeilen eine nach dem anderen herausgelesen, in der Reihenfolge, in der sie stehen. Ist der letzte Wert aus einer DATA-Zeile herausgelesen, so wird, wenn vorhanden und abgefragt, der erste Wert aus der nächsten DATA-Zeile genommen.

Die erste Form des RESTORE-Befehls setzt den Zeiger an den Anfang der ersten DATA-Zeile zurück.

Die zweite Form setzt den Zeiger an den Anfang der ersten DATA-Zeile, deren Zeilennummer größer oder gleich der durch ausdruck angegebenen Zeilennummer ist.

**Beispiel:**

10: DIM B(10)	Dimensionierung eines Feldes
20: WAIT 32	
30: FOR I=1 TO 10	Setzt den Zeiger an den Anfang der 1. DATA-Zeile.
40: RESTORE	Den Variablen von B werden die Werte zugewiesen.
50: READ B(I)	Alle Elemente haben den Wert 20.
60: PRINT B(I) * I;	
70: NEXT I	
80: DATA 20	
90: END	

## RETURN

### 1. RETURN

Abkürzung: RE., RET., RETU., RETUR.

Vergleiche: GOSUB, ON...GOSUB

#### Wirkung:

Mit dem RETURN-Befehl wird eine Unterroutine geschlossen, der Programmablauf wird mit dem Befehl, der auf den GOSUB-Befehl folgt, fortgesetzt.

#### Anwendung:

Eine Unterroutine kann mehr als ein RETURN enthalten. Erreicht das Programm einen dieser RETURN-Befehle, wird die Unterroutine geschlossen und das Programm mit dem Befehl hinter dem GOSUB bzw. ON...GOSUB, von dem der Rechner in die Unterroutine gesprungen ist, fortgesetzt. Wird ein RETURN ohne GOSUB ausgeführt, erhalten Sie die Fehlermeldung ERROR 5.

#### Beispiel:

10: GOSUB 100	Das Programm druckt "HALLO" aus und springt
20: END	zurück in Zeile 20.
100: PRINT "HALLO"	
110: RETURN	

**STOP****1. STOP**

Abkürzung: S., ST., STO.

Vergleiche: END; CONT-Kommando

STOP

STOP

**Wirkung:**

Mit dem STOP-Befehl wird die Ausführung eines Programms gestoppt. Die Ausführung kann durch den Befehl CONT wieder aufgenommen werden.

**Anwendung:**

Erreicht der Computer in einem Programm einen STOP-Befehl, so wird die Ausführung des Programms abgebrochen und Sie erhalten eine Meldung wie z.B. "BREAK IN 200", wenn das Programm in Zeile 200 gestoppt wurde. Sie können jetzt die Variablen des Programms überprüfen und so die Ausführung desselben kontrollieren. Die Ausführung des Programms kann mit dem CONT-Befehl fortgesetzt werden.

**Beispiel:**

10: STOP : BREAK IN 10 erscheint auf der Anzeige.

**TROFF****1. TROFF**

Abkürzung: TROF.

Vergleiche: TRON

**Wirkung:**

Mit dem TROFF-Befehl wird der Trace-Modus ausgeschaltet.

**Anwendung:**

Die Ausführung des Befehls TROFF hat zu Folge, daß ein Programm wieder auf "normale" Weise ausgeführt wird.

**Beispiel:**

10: TRON	Bei Ausführung des Programms werden die Zeilen-
20: FOR I=1 TO 3	nummern ausgegeben:
30: NEXT I	10, 20, 30, 30 und 40
40: TROFF	

## TRON

### 1. TRON

Abkürzung: TR., TRO.

Vergleiche: TROFF

#### Wirkung:

Der TRON-Befehl schaltet den Trace-Modus ein.

#### Anwendung:

Der Trace-Modus unterstützt die Fehlersuche in Programmen. Ist der Trace-Modus eingeschaltet, so wird die Zeilennummer der gerade ausgeführten Zeile auf der Anzeige ausgegeben. Der Computer wartet dann auf den Runter-Pfeil, um die nächste Zeile auszuführen. Mit dem Hoch-Pfeil kann die gerade ausgeführte Zeile zur Anzeige gebracht werden. Der Trace-Modus bleibt so lange eingeschaltet, bis er mit dem TROFF-Befehl wieder ausgeschaltet wird oder bis Sie die Tastenfolge **[SHIFT] [C-CE]** eingeben.

#### Beispiel:

10: TRON

Bei Ausführung des Programms werden die Zeilennummern ausgegeben:

20: FOR I=1 TO 3

10, 20, 30, 30, 30 und 40

30: NEXT I

40: TROFF

## USING

### 1. USING

### 2. USING "spezifikation"

Abkürzung: U., US., USI., USIN.

Vergleiche: LPRINT, PAUSE, PRINT

Weitere Erklärung zu USING finden Sie im Anhang C.

#### Wirkung:

Mit dem USING-Befehl kann das Anzeigenfeld oder der Druck formatiert werden.

#### Anwendung:

Der USING-Befehl kann einzeln oder als Erweiterung der Befehle LPRINT, PAUSE oder PRINT eingesetzt werden. Der USING-Befehl erstellt eine Ausgabeformatierung, die für alle folgenden Ausgabefehle Gültigkeit hat, bis die Formatierung durch einen neuen USING-Befehl geändert wird.

Die Spezifikationen des Befehls bestehen aus einer in Anführungsstrichen ("") gesetzten Zeichenkette. Die Kette ist aus folgenden Zeichen zusammengesetzt:

- # Rechtsbündiges Feld für numerische Zeichen
- .
- Dezimalpunkt
- ^ Zahlen sollen in Exponentialschreibweise wiedergegeben werden.
- & Linksbündiges Feld für alphanumerische Zeichen

Die Spezifikation "#####" erstellt z. B. ein rechtsbündiges Feld für numerische Zeichen mit Platz für drei Ziffern und das Vorzeichen. In numerischen Feldern muß immer eine Position für das Vorzeichen vorgesehen sein, auch, wenn nur positive Zahlen ausgegeben werden sollen.

Spezifikationen können mehr als ein Feld beschreiben. "#####&&&&" setzt z. B. ein numerisches Feld und ein alphanumerisches Feld direkt nebeneinander.

Wird, wie in Form 1 des Befehls, keine Spezifikation gegeben, so werden alle gesetzten Formate gelöscht und die Standardformatierung tritt wieder in Kraft.

**Beispiele:**

10: A=125 : X\$="ABCDEF"

**ANZEIGE**

20: PRINT USING "##.##^"; A

1.25E02

30: PRINT USING "&amp;&amp;&amp;&amp;&amp;&amp;"; X\$

ABCDEF

40: PRINT USING "####&amp;&amp;&amp;"; A; X\$

125ABC

## WAIT

1. WAIT
2. WAIT ausdruck

Abkürzung: W., WA., WAI.

Vergleiche: PRINT

### Wirkung:

Der WAIT-Befehl setzt die Zeit fest, die eine Ausgabe auf der Anzeige wieder-gegeben werden soll.

### Anwendung:

Bei normaler Ausführung eines Programms wartet der Computer nach einem PRINT-Befehl, bis die **[ENTER]**-Taste gedrückt wird. Der WAIT-Befehl weist den Computer nun an, eine Ausgabe für eine vorgegebene Zeit anzuzeigen und dann die Ausführung des Programms fortzusetzen, genau wie mit dem PAUSE-Befehl. Der ausdruck im WAIT-Befehl setzt die Länge des Intervalls fest. Er kann jeden Wert von 0 bis 65535 haben. Dabei bedeutet die Erhöhung um 1 eine Verlängerung des Intervalls um 1/59 Sekunde. WAIT ist zu schnell, als daß die Ausgabe gelesen werden könnte; WAIT 65535 setzt das Intervall auf ca. 19 Minuten. WAIT ohne Ausdruck setzt den Computer auf seine normale Arbeitsweise zurück.

### Beispiel:

10: WAIT 59

Der Rechner wartet nach PRINT etwa 1 Sekunde.

## 2.3.3 FUNKTIONEN

### 2.3.3.1 PSEUDO VARIABLE

Pseudovariablen sind eine Gruppe von Funktionen, die ohne Argument wie einfache Variable behandelt werden.

#### INKEY\$

##### 1. INKEY\$

INKEY\$ ist eine String-Pseudovariable, die den Wert der letzten, auf der Tastatur gedrückten Taste zugewiesen bekommt. Die Tasten **ENTER**, **C-CE**, **SHIFT**, **DEF**, **SML**, **↑ -Pfeil**, **↓ -Pfeil**, **→ -Pfeil**, **CAL** und **BASIC** haben den Wert NUL (= leer). Mit INKEY\$ können Eingaben über die Tastatur ohne ein abschließendes **ENTER** gemacht werden.

##### 5: WAIT 50

##### 10: A\$=INKEY\$

##### 20: B=ASC A\$

##### 30: IF B=0 THEN GOTO 10

##### 40: IF B=... THEN ...

Die Zeilen ab Zeile 40 können eine Tastatur-Testroutine enthalten (z. B. 40 PRINT A\$). Wenn das Programm gestartet wird, ist der Wert von INKEY\$ = NUL, da die letzte gedrückte Taste **ENTER** war. Folgt INKEY\$ auf einen PRINT- oder PAUSE-Befehl, so nimmt INKEY\$ den Wert der Anzeige an.

- Wenn sich am Anfang des Programms ein INKEY\$ Befehl befindet, kann es vorkommen, daß beim Starten des Programms die Starttaste vom INKEY\$ Befehl gelesen wird. Beispielsweise kann es im folgenden Programm  
 10 "Z": Z\$ = INKEY&  
 vorkommen, daß die Taste **Z** gelesen wird, wenn das Programm durch Drücken der Tasten **DEF** **Z** gestartet wird.

**MEM****1. MEM**

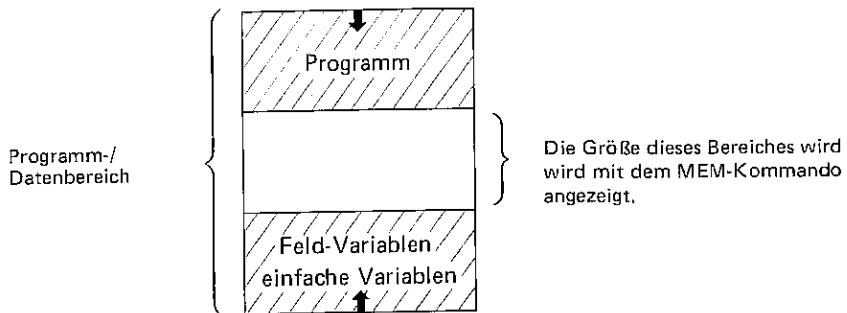
Abkürzung: M.

**Wirkung:**

Zum Erhalten der Anzahl der freien Bytes im Programm/Daten-Bereich.

**Anwendung:**

Liefert die Anzahl der freien Bytes (der nicht durch ein Programm, Feldvariable oder einfache Variable belegte Bereich) im Programm/Daten-Bereich.



Die Programmgröße (in Bytes) kann durch folgende Eingabe angezeigt werden:

Beispiel: Computer mit RAM-Karte CE-211M

**RUN-Modus**

**CLEAR [ENTER]**

(löscht die einfachen Variablen, die Feld-Variablen usw.)

**3070-MEM [ENTER]**

(Es wird die Programmgröße  
– in Bytes – angezeigt.)



Eingabe der gesamten RAM-Speichergröße (in Bytes).

**RAM-Speichergröße (in Bytes)**

Nur mit CE-211M	Nur mit CE-201M	Nur mit CE-202M
3070	7166	15358

**PI****1. PI**

PI ist eine numerische Pseudovariable, die den Wert  $\pi$  hat. Sie hat die gleiche Funktion wie die spezielle Taste  $\pi$  auf der Tastatur. Wie alle anderen Zahlen wird auch der Wert von PI mit einer Genauigkeit von 10 Stellen wiedergegeben (3.141592654).

### 2.3.3.2 NUMERISCHE FUNKTIONEN

Numerische Funktionen sind eine Gruppe von mathematischen Funktionen, die aus einem numerischen Argument ein numerisches Ergebnis errechnen. Die Gruppe enthält trigonometrische und logarithmische Funktionen, sowie solche, die ganze Zahlen und Beträge errechnen bzw. Vorzeichen ermitteln. Viele BASIC-Dialekte fordern, daß das Argument der Funktion in Klammern steht. Dieser Computer braucht Klammern nur dann, wenn in einer komplexen Berechnung das Argument der Funktion genau abgegrenzt werden muß.

LOG 100+100 wird als

(LOG 100) + 100 und nicht als LOG (100 + 100)

interpretiert.

#### ABS

##### 1. ABS numerischer ausdruck

Errechnet den Absolut-Betrag des angegebenen Arguments.

ABS -10 ist gleich 10

#### ACS

##### 1. ACS numerischer ausdruck

Errechnet den Arcuscosinus des Arguments. Arcuscosinus ist die Umkehrfunktion des Cosinus. Das Ergebnis ist also der Winkel zu dem angegebenen Wert. Die Darstellung des Ergebnisses ist abhängig vom Winkelmodus des Computers (DEG, RAD, GRAD).

ACS .5 ist im DEG-Modus gleiche 60

#### AHC

##### 1. AHC numerischer ausdruck

Errechnet den Areacosinus des angegebenen Arguments.

AHC5 ist gleich 2.29243167.

**AHS****1. AHS numerischer ausdruck**

Errechnet den Areasinus des angegebenen Arguments.

AHS6 ist gleich 2.491779853.

**AHT****1. AHT numerischer ausdruck**

Errechnet den Areatangens des angegebenen Arguments.

**ASN****1. ASN numerischer ausdruck**

Errechnet den Arcussinus des angegebenen Argumentes. Arcussinus ist die Umkehrfunktion des Sinus. Die Darstellung des Ergebnisses ist abhängig vom Winkelmodus des Computers (DEG, RAD, GRAD).

ASN .5 ist im DEG-Modus gleich 30

**ATN****1. ATN numerischer ausdruck**

Das Ergebnis ist der Arcustangens des Arguments. Arcustangens ist die Umkehrfunktion des Tangens. Die Darstellung des Ergebnisses ist abhängig vom Winkelmodus des Computers (DEG, RAD, GRAD).

ATN 1 ist im DEG-Modus gleich 45.

**COS****1. COS numerischer ausdruck**

Errechnet den Cosinus eines Winkels. Die Darstellung des Ergebnisses ist abhängig vom Winkelmodus des Computers (DEG, RAD, GRAD).

COS 60 ist im DEG-Modus gleich .5

**CUR****1. CUR numerischer ausdruck**

Errechnet die Kubikwurzel des gegebenen Arguments.

CUR8 ist gleich 2.

**DEG****1. DEG numerischer ausdruck**

Das Ergebnis ist die Umwandlung des Winkelarguments (Alt-Grad) als DMS-Darstellung (Grad, Minute, Sekunde) in die Dezimaldarstellung DEG. Im DMS-Format stellt der ganzzahlige Anteil der Zahl die Grandzahl, die ersten beiden Dezimalstellen die Minuten und die dritte und vierte Dezimalstelle die Sekunden dar. Die weiteren Stellen geben die Dezimalsekunden wieder. 55 Grad 10' 44,5'' warden also z.B. als 55.10445 dargestellt. Im DEG-Format ist der ganzzahlige Anteil wieder Grad, aber die Dezimalstellen sind Dezimalgradbruchteile.

DEG 55.10445 ist also gleich 55.17902778.

**DMS****1. DMS numerischer ausdruck**

DMS ist die Umkehrfunktion zur DEG-Funktion (vergleiche DEG).

DMS 55.17902778 ist gleich 55.10445.

**EXP****1. EXP numerischer ausdruck**

Errechnet die Potenzen der Zahl e (= 2.718181828, die Basis des Logarithmus naturalis).

EXP 1 ist gleich 2.718291829

**FACT****1. FACT numerischer ausdruck**

Errechnet die Fakultät des gegebenen Arguments.

FACT5 ist gleich 120.

**HCS****1. HCS numerischer ausdruck**

Errechnet den Hyperbelcosinus des gegebenen Arguments.

HCS5 ist gleich 74.20994852.

**HSN****1. HSN numerischer ausdruck**

Errechnet den Hyperbelsinus des gegebenen Arguments.

HSN4 ist gleich 27.2899172

**HTN****1. HTN numerischer ausdruck**

Errechnet den Hyperbeltangens des gegebenen Arguments.

HTN1 ist gleich 0.761594156.

**INT****1. INT numerischer ausdruck**

Das Ergebnis ist der ganzzahlige Anteil des Arguments.

INT PI ist gleich 3.

**LN****1. LN numerischer ausdruck**

Errechnet den Logarithmus zur Basis e (2.718281828) des Arguments.

LN 100 ist gleich 4.605170186

**LOG****1. LOG numerischer ausdruck**

Errechnet den Logarithmus zur Basis 10 des Arguments.

LOG 100 ist gleich 2.

**POL****1. POL (numerischer ausdruck, numerischer ausdruck)**

Bewirkt die Umwandlung von rechtwinkligen Koordinaten in Polarkoordinaten. Der erste numerische Ausdruck ist die Entfernung von der Y-Achse und der zweite die Entfernung von der X-Achse. Die berechneten Werte für die Entfernung und den Winkel der Polarkoordinaten werden den festen Variablen Y bzw. Z zugeordnet. Der Wert des Winkels ist dabei von der Winkeleinheit (DEGREE, GRAD, RADIANT) abhängig.

POL (3, 4) ist gleich (5,53.13010235) in DEGREE.

**RCP****1. RCP numerischer ausdruck**

Errechnet den Reziprokwert des gegebenen Arguments.

RCP5 ist gleich 0.2.

**REC****1. REC (numerischer ausdruck, numerischer ausdruck)**

Bewirkt die Umwandlung von Polarkoordinaten in rechtwinklige Koordinaten. Der erste numerische Ausdruck ist die Entfernung und der zweite der Winkel, der von der Winkeleinheit (DEGREE, GRAD, RADIANT) abhängig ist. Die berechneten Werte, die Entferungen von der Y-Achse und von der X-Achse, werden den festen Variablen Y bzw. Z zugeordnet.

REC (7, 50) ist gleich (4.499513268, 5.362311102) in DEGREE.

**RND****1. RND numerischer ausdruck**

Erzeugt Zufallszahlen. Ist das Argument kleiner 1, aber größer 0, so ist das Ergebnis kleiner 1 und größer oder gleich 0. Ist das Argument eine ganze Zahl und größer oder gleich 1, so ist das Ergebnis kleiner oder gleich dem Argument und größer oder gleich 1. Ist das Argument keine ganze Zahl und größer 1, so ist das Ergebnis kleiner oder gleich der nächstgrößeren ganzen Zahl des Arguments und größer oder gleich 1.

**Argument****Ergebnis**

TYP

<b>Untergrenze</b>	<b>Obergrenze</b>
.5	<1
2	2
2.5	3

Die Sequenz ist jedes Mal die gleiche, wenn der Computer eingeschaltet wird, da die Basiszahl immer die gleiche ist. Um diese Basis-Zahl zufällig zu ändern, brauchen Sie den RANDOM-Befehl.

**ROT****1. numerischer ausdruck ROT numerischer ausdruck**

Errechnet die Potenzwurzel des angegebenen Arguments.

125ROT3 ist gleich 5.

(Beispiel:  $\sqrt[3]{125}$  muß als 125 ROT 3 eingegeben werden.)

**SGN****1. SGN numerischer ausdruck**

Das Ergebnis ist abhängig vom Vorzeichen des Arguments. Ist das Argument positiv, ist das Ergebnis gleich 1; ist das Argument negativ, so ist das Ergebnis gleich -1; wenn das Argument gleich 0 ist, ist auch das Ergebnis gleich 0.

SGN -5 ist gleich -1.

**SIN****1. SIN numerischer ausdruck**

Errechnet den Sinus des Winkelarguments. Die Darstellung ist abhängig von dem Winkelmodus des Computers (DEG, RAD, GRAD).

SIN 30 ist im DEG-Modus gleich .5.

**SQR****1. SQR numerischer ausdruck**

Errechnet die Quadratwurzel des Arguments. Das Ergebnis ist identisch mit der Anwendung der speziellen -Taste der Tastatur.

SQR 4 ist gleich 2.

**SQU****1. SQU numerischer ausdruck**

Errechnet das Quadrat des gegebenen Arguments.

SQU3 ist gleich 9.

**TAN****1. TAN numerischer ausdruck**

Errechnet den Tangens des Winkelarguments. Die Darstellung ist abhängig vom Winkelmodus des Computer (DEG, RAD, GRAD).

TAN 45 ist im DEG-Modus gleich 1.

**TEN****1. TEN numerischer ausdruck**

Exponentialfunktion mit Basis 10 und numerischer Ausdruck als Exponent.

TEN3 ist gleich 1000.

### 2.3.3.3 STRING-FUNKTIONEN

Ein String ist eine Kette aus alphanumerischen und Sonderzeichen. String-Funktionen sind eine Gruppe von Operationen, die Strings manipulieren. Sie nehmen einen String und errechnen einen numerischen Wert dazu, oder Sie nehmen einen numerischen Wert und errechnen den String dazu, oder Sie bearbeiten den String als String. In vielen BASIC-Dialekt en muß das Argument der Funktion in Klammern gesetzt werden. Nicht so bei diesem Computer. Klammern müssen nur dann gesetzt werden, wenn es nötig ist, das Argument von anderen Werten deutlich zu trennen. String-Ausdrücke mit zwei oder drei Ausdrücken benötigen immer eine Klammerung.

#### **ASC**

##### 1. ASC "string ausdruck"

Errechnet den Zeichen-Code des ersten Zeichens des Ausdrucks. Die Zeichen-Code-Tabelle finden Sie in Anhang B.

ASC "A" ist gleich 65.

Der Computer benutzt ASCII-Codes und ihre Zeichen.

#### **CHR\$**

##### 1. CHR\$ numerischer ausdruck

Errechnet das Zeichen des gegebenen numerischen Wertes. CHR\$ ist die Umkehrfunktion der ASC-Funktion.

CHR\$ 65 ist gleich "A".

**Achtung:** Wenn der Zeichencode von 13 angegeben wird, während Sie manuell das CHR\$-Kommando ausführen, wird der jeweils folgende Inhalt nicht angezeigt.

Beispiel:

CHR\$70+CHR\$71+CHR\$13+CHR\$75+CHR\$76

**ENTER** → FG

Die Zeichen K und L für die Codes 75 und 76 werden nicht angezeigt.

## LEFT\$

### 1. LEFT\$ ("string ausdruck", numerischer ausdruck)

Diese Funktion erstellt einen Teil-String des durch string ausdruck angegebenen Strings. Von links beginnend wird die mit numerischer ausdruck gegebene Zahl von Zeichen herausgenommen.

LEFT\$ ("ABCDEF", 2) ist gleich "AB".

## LEN

### 1. LEN "string ausdruck"

Errechnet die Länge eines Strings.

LEN "ABCDE" ist gleich 5.

## MID\$

### 1. MID\$ ("string ausdruck", num asdr1, num asdr2)

Diese Funktion erstellt einen Teil-String aus dem durch ausdruck angegebenen string. Durch num asdr1 wird das erste Zeichen bestimmt, num asdr2 bestimmt die Anzahl der herauszunehmenden Zeichen.

MID\$ ("ABCDEF", 2, 3) ist gleich "BCD".

## RIGHT\$

### 1. RIGHT\$ ("string ausdruck", numerischer ausdruck)

Diese Funktion erstellt einen Teil-String des durch string ausdruck angegebenen Strings. Von rechts beginnend wird die Anzahl der von numerischer ausdruck gegebenen Zeichen herausgenommen.

RIGHT\$ ("ABCDEF", 3) ist gleich "DEF".

## STR\$

### 1. STR\$ numerischer ausdruck

STR\$ wandelt einen numerischen Wert in einen String um.

STR\$ 1.59 ist gleich "1.59".

**VAL****1. VAL "string ausdruck"**

VAL ist die Umkehrfunktion zu STR\$. VAL wandelt einen String in einen numerischen Ausdruck um.

VAL "1.59" ist gleich 1.59.

**Achtung:** VAL kann nur numerische Zeichen (0 bis 9), Vorzeichen (+ und -) und das "E" für Exponenten umwandeln! Sind andere Zeichen in dem string ausdruck, so werden alle Zeichen rechts des letzten erlaubten Zeichens ignoriert. Spaces werden wie nicht-existent behandelt.

VAL "1.59 abc" ist gleich 1.59

VAL

VAL "1.59 abc" ist gleich 1.59

VAL "1.59 abc" ist gleich 1.59

VAL "1.59 abc" ist gleich 1.59

VAL

VAL "1.59 abc" ist gleich 1.59

VAL "1.59 abc" ist gleich 1.59

VAL

VAL "1.59 abc" ist gleich 1.59

VAL "1.59 abc" ist gleich 1.59

VAL "1.59 abc" ist gleich 1.59

### **2.3.4 EIN-/AUSGABE-KOMMANDOS**

#### **CLOSE**

##### **1. CLOSE**

Abkürzung: CLOS.

Vergleiche: OPEN

#### **Wirkung:**

Schließt die Leitung zum seriellen Interface.

#### **Anwendung:**

Das Kommando CLOSE schließt die Leitung, die für Ein-/Ausgabe zu seriellen Interface mit OPEN geöffnet wurde.

Nach diesem Kommando können also Daten über das Interface weder empfangen noch gesendet werden.

## CONSOLE

### 1. CONSOLE ausdruck

Abkürzung: CONS., CONSO., CONSOL.

Vergleiche: OPEN, LPRINT, LLIST

#### Wirkung:

Setzt bei Datensendung die Zahl der Zeichen pro Zeile.

#### Anwendung:

CONSOLE setzt für die Kommandos LLIST und LPRINT die Zahl der Zeichen pro Zeile fest, wenn Daten über das serielle Interface gesendet werden.

\* Der Wert für ausdruck muß ganzzahlig sein und darf nicht kleiner als 1 oder größer als 160 sein. Wird ausdruck größer 160 gesetzt, so wird die Zahl der Zeichen pro Zeile auf 160 begrenzt.

Wird ausdruck kleiner 1 gesetzt, erhalten Sie die Fehlermeldung ERROR 3.

Wenn Sie im Kommando CONSOLE keinen Wert angeben, wird das Kommando ignoriert und die alte Zahl der Zeichen pro Zeile benutzt.

\* Der voreingestellte Wert ist 39 Zeichen pro Zeile, der auch wieder eingenommen wird, wenn die Batterien des Rechners gewechselt werden.

**INPUT # 1**

1. INPUT# 1 variable, variable, variable, variable, . . .

Abkürzung: I.#1, IN.#1, INP.#1, INPU.#1

Vergleiche: OPEN, PRINT#1

**Wirkung:**

Lädt Daten über das serielle Interface in den Computer und weist sie den definierten Variablen zu.

**Anwendung:**

- \* Das Kommando wird nur ausgeführt, wenn die Leitung zum seriellen Ein-/Ausgabe-Interface vorher mit dem OPEN-Kommando geöffnet wurde. Ansonsten wird das Kommando ignoriert.
- \* Das INPUT# 1-Kommando erwartet Daten und weist sie den angegebenen Variablen zu. Die Variablen müssen also dem Datentyp entsprechen. Vergleichen Sie hierzu mit dem PRINT#1-Kommando.

**Beispiel:**

INPUT# 1A, AB, C\$, E(\*)

Die empfangenen Daten werden den Variablen A, AB, C\$ und dem Feld E( ) zugewiesen.

- \* Achten Sie darauf, daß die empfangenen Daten mit dem Variablentyp übereinstimmen, also Zahlen numerischen Variablen und Strings String-Variablen zugewiesen werden.

Wird im ASCII-Codesystem ein Zeichen einer numerischen Variablen zugewiesen, so wird ihr Wert gleich null gesetzt. Wird eine Zahl einer String-Variablen zugewiesen, so wird die Zahl in einen String umgewandelt. Sind die Daten und die Variablen nicht gleichen Typs, werden Sie also ungewollte Werte auf Ihren Variablen führen.

Selbst, wenn Daten in der Form einer Funktion, wie z.B. "SIN30", einer numerischen Variablen zugewiesen werden, wird die Variable gleich null gesetzt.

Bei Daten der Form "10+40" werden Zeichen nach dem Operationssymbol ignoriert, in diesem Fall wird nur "10" erfaßt.

#### **Beachten Sie:**

1. Enthalten die Engabedaten "CR" (Kontrollcode 0DH) oder "NULL" (Ø), können alle diesem Zeichen folgenden Daten ignoriert werden.
  2. Einfache und Feld-Variable müssen im Programm-/Datenspeicherbereich Raum zugewiesen bekommen, bevor das INPUT#1-Kommando ausgeführt wird. Sonst bekommen Sie eine Fehlermeldung.

After the first year, participants will receive a monthly stipend of \$1000, plus a \$1000 bonus if they complete all three years.

**LLIST**

1. LLIST
2. LLIST A
3. LLIST ausdruck1, ausdruck2

A "label" oder ausdruck

Abkürzung: LL., LLI., LLIS.

Vergleiche: OPEN, CONSOLE

**Wirkung:**

Sendet Programm-Listings über das serielle Interface.

**Anwendung:**

Dans LLIST-Kommando kann sowohl im PROGRAMM-Modus als auch im RUN-Modus eingesetzt werden.

Ist das Ein-/Ausgabe-Interface vorher mit dem OPEN-Kommando geöffnet worden, sendet das LLIST-Kommando das gespeicherte Programm als Listing im ASCII-Code an angeschlossene Geräte.

Ist die Leitung zum Interface geschlossen, wird das Listing über den Drucker ausgegeben.

\* Form 1 sendet alle gespeicherten Programm-Listings.

**Beispiel:**

Geben Sie folgendes Demoprogramm ein:

```
10 OPEN  
100 REM **ABC-12**  
65279 END
```

Geben Sie dann das Kommando:

LLIST **ENTER**

Das Listing wird auf dem Drucker ausgegeben. Geben Sie jetzt das Kommando:

```
OPEN ENTER  
LLIST ENTER
```

Das Listing wird jetzt über das Interface ausgegeben.

- \* Form 2 sendet nur die Zeile, die mit dem Wert von ausdruck oder dem gegebenen "label" gekennzeichnet ist.
- \* Form 3 sendet alle Zeilen, die zwischen den mit ausdruck1 und ausdruck2 benannten Zeilen liegen (inklusiv). ausdruck1 und ausdruck2 können auch durch "label" ersetzt werden.
- \* Wird in Form 3 ausdruck1 weggelassen, so werden alle Zeilen von der ersten bis zu der mit ausdruck2 bezeichneten Zeile ausgegeben.
- \* Wird in Form 3 ausdruck1 weggelassen, so werden alle Zeilen ab der mit ausdruck1 bezeichneten Zeile bis zu letzten ausgegeben.
- \* Gibt es keine Zeile mit der Nummer des ausdruck1 oder ausdruck2, so wird jeweils die nächsthöhere Zeile genommen. Ist die Zeile von ausdruck1 und ausdruck2 die gleiche, erhalten Sie die Fehlermeldung ERROR 1.
- \* Wurde ein Passwort gesetzt, wird das Kommando LLIST ignoriert.
- \* Wurden weitere Programme mit MERGE eingeladen, so wird mit LLIST nur das letzte geladene Programm ausgegeben. Wollen Sie die vorher geladenen Programme anzeigen lassen, müssen Sie das Kommando in dieser Form benutzen: LLIST "label"
- \* Die Zahl der Zeichen pro Zeile wird mit dem CONSOLE-Kommando festgelegt. Wurde diese Zahl kleiner oder gleich 23 gesetzt, erhalten Sie die Fehlermeldung ERROR 3.
- \* Vor der Aussendung wird "π" in PI und "√" in SQR umgewandelt.

## LOAD

### 1. LOAD

Abkürzung: LOA.

Vergleiche: OPEN, CLOAD

#### Wirkung:

Lädt Daten über das serielle Ein-/Ausgabe-Interface in den Programm-/Datenspeicherbereich.

#### Anwendung:

Das Kommando wird nur ausgeführt, wenn die Leitung zum seriellen Interface vorher mit dem OPEN-Kommando geöffnet wurde. Ansonsten wird es ignoriert.

\* Der Rechner lädt so lange Daten über das Interface ein, bis der Endcode erreicht ist. Dieser Datenblock wird als eine Programmzeile interpretiert. Der Computer wandelt die Daten dann in eine Form um, die er als Programm in seinem Programm-/Datenspeicherbereich ablegen kann. Dann liest er den nächsten Datensatz bis zum nächsten Endcode ein.

Dieser Vorgang wiederholt sich, bis der Textendecode erreicht ist. Vergleichen Sie dazu mit dem OPEN-Kommando.

\* In einem Datensatz können bis zu 256 Byte gelesen werden. Sollen also mehr als 256 Byte empfangen werden, erhalten Sie eine Fehlermeldung.

\* Die eingelesenen Daten werden umgeformt und im Speicher abgelegt. Ist eine Programmzeile inklusiv Zeilennummer länger als 80 Byte, so erhalten Sie eine Fehlermeldung.

Außerdem muß der Anfang einer Zeile immer ein numerischer Wert, also eine Zeilennummer, sein.

\* Während der Ausführung des LOAD-Kommandos werden die Zeilen nicht neu sortiert (in aufsteigender Folge).

#### Beachten Sie:

\* Die Ausführung des LOAD-Kommandos endet, wenn der Textendecode gelesen wird.

Wird der Code nicht gesendet, müssen Sie die Ausführung wie folgt beenden:

1. Senden Sie nach der beendeten Übertragung des Programmes vom sendenden Gerät einen Textendecode.

2. Oder drücken Sie die **BRK**-Taste, um die Übertragung zu beenden.

## Zwischencodes

- \* Daten werden als Zwischencodes gelesen und in den Programm/Daten-Bereich geschrieben.
  - \* Wenn die Ausführung von LOAD aufgrund eines Fehlers unterbrochen wird, ist das gesamte Programm gelöscht. (Variable bleiben erhalten.)

After the first few days, the new system worked well, but as time went on, the system became increasingly unreliable, with frequent errors and downtime.

As a result of the above-mentioned, the author has come to the conclusion that the main factor in the formation of the new type of society is the development of the spiritual culture of the people. The author believes that the spiritual culture of the people is the most important factor in the formation of the new type of society. The author also believes that the spiritual culture of the people is the most important factor in the formation of the new type of society.

After the initial period of adjustment, the economy stabilizes at a new equilibrium level of output. The shift from the initial equilibrium to the new equilibrium is labeled as a "reallocation of economic resources".

Figure 10. A 3D visualization of a real-world dataset showing the effect of different parameter settings on the learned model.

<sup>10</sup> See, e.g., *U.S. v. Babbitt*, 100 F.3d 1250, 1256 (10th Cir. 1996) (“[T]he [Bald Eagle] Act does not prohibit the killing of bald eagles.”); *U.S. v. Ladd*, 100 F.3d 1250, 1256 (10th Cir. 1996) (“[T]he [Bald Eagle] Act does not prohibit the killing of bald eagles.”).

**LPRINT**

1. LPRINT A
2. LPRINT A,A,...,A
3. LPRINT A;A;...;A
4. LPRINT ... A;      Erweiterung der Formen 1, 2, 3 um ein Semikolon am Ende des Kommandos

**5. LPRINT**

A                ausdruck oder string

Abkürzung: LP., LPR., LPRI., LPRIN.

Vergleiche: OPEN, CONSOLE, USING

**Wirkung:**

Sendet Daten über das serielle Ein-/Ausgabe-Interface.

**Anwendung:**

Wurde die Leitung zum seriellen Interface mit dem OPEN-Kommando geöffnet, so werden die Daten im ASCII-Code übertragen. Ist die Leitung geschlossen, werden die Daten auf dem Drucker ausgegeben.

- \* Form 1 sendet den Wert des Ausdrucks oder die Zeichenkette (String). Ist der Wert des Ausdrucks negativ, wird ihm ein Minuszeichen (-) vorangestellt. Ist er positiv, ein Leerfeld.
- \* Form 2 setzt automatisch 12 Zeichen pro Einheit fest und sendet die Daten in dieser Form.

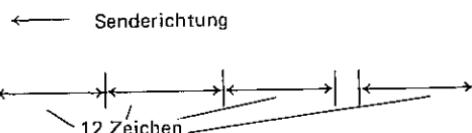
**Beispiel:**

10 OPEN "1200, N, 8, 1, A, C"

20 CONSOLE 36

30 LPRINT 12345, "ABCDE", -7/5, 1.23456789E12

Bei Ausführung werden die Daten in folgender Form gesendet:



Die Daten werden in 12-Zeichen-Blöcken gesendet. Der erste Block reicht bis zum Punkt(.) vor dem "A", der zweite bis zum Punkt(.) vor dem Minuszeichen(-), der dritte bis zur "4" vor dem "CR", der vierte ab dem Punkt(.) vor der "1" bis zur "2" vor dem zweiten "CR".

Das erste "CR" (Carriage Return – Zeilenende) wurde zwischengeschoben, da die Zeilenlänge mit dem Kommando CONSOLE 36 auf 36 Zeichen festgesetzt wurde. Das zweite "CR" kennzeichnet das Ende des Datensatzes.

Wenn ein String länger als 12 Zeichen ist, so werden nur die ersten 12 Zeichen des Strings gesendet.

Ebenso, wenn der Wert eines Ausdrucks mehr als 12 Stellen hat. Die Zahl wird in Exponentialdarstellung übertragen, die niederwertigen Stellen hinter dem Dezimalpunkt werden nicht gesendet. Vor der Zahl wird immer ein Leerfeld gesendet, wenn die Zahl positiv ist. Ist sie negativ, wird ein Minuszeichen vorangestellt.

- \* Form 3 verändert nicht die Länge der zu sendenden Daten. Einer positiven Zahl wird kein Leerfeld vorangestellt.

#### Beispiel:

50 LPRINT -123; "ABC"; 567.89

Gesendet wird:

-123.ABC567.89CR

Der Datensatz wird, je nach OPEN-Modus, mit "CR" (Carriage Return – Wagenrücklauf) oder "LF" (Line Feed – Zeilenvorschub) beendet.

- \* Form 4 sendet keinen Datensatzendcode. Es wird nur der durch das CONSOLE-Kommando gesetzte "CR"-Code übertragen.
- \* Form 5 sendet nur den Datensatzendcode.
- \* Wurde das Format mit dem USING-Kommando vorgegeben, werden die Daten in allen Formen des Kommandos LPRINT entsprechend diesem USING übertragen.
- \* Vor der Aussendung wird " $\sqrt{ }$ " in SQR und " $\pi$ " in PI umgewandelt.
- \* Die Ausführung von PRINT=LPRINT lässt den Rechner alle PRINT-Kommandos als LPRINT-Kommandos interpretieren. Das Kommando PRINT = LPRINT ist nur erlaubt, wenn der Drucker angeschlossen oder die Leitung zum seriellen Ein-/Ausgabe-Interface geöffnet ist.

#### Achtung:

Um Zeichen oder Kontrollcodes zu senden, die nicht auf der Tastatur vorhanden sind, müssen Sie das CHR\$-Kommando benutzen.

Sie wollen das Technische AT (@) senden:

(1) .  
. .  
50 LPRINT CHR\$ &40

(2) .  
. .  
50 A\$=CHR\$&40  
60 LPRINT A\$

Der Kontrollcode "NULL" (Ø) darf nur in Form 1 gesendet werden. In Form 2 wird er ignoriert.

**OPEN**

1. OPEN "baud rate, parität, wortlänge, stop bit, code typ, end code, textende code"
2. OPEN

Abkürzung: OP., OPE.

Vergleiche: CLOSE

**Wirkung:**

Öffnet die Leitung zum seriellen Ein-/Ausgabe-Interface und ermöglicht somit den Datentransfer über das Interface, setzt die dafür nötigen Optionen.

**Anwendung:**

Beide Formate öffnen die Leitung zum Interface.

- \* Form 2 benutzt die Öffnungsoptionen eines früher gegebenen OPEN-Kommandos. Wurden die Batterien gewechselt, oder wird das erste Mal das OPEN-Kommando gegeben, so werden die Standardwerte eingesetzt.
- \* Form 1 erlaubt die Änderung der Öffnungsoptionen. Folgende Optionen können gesetzt werden:

"baud rate, parität, wortlänge, stop bit, code typ, end code, textende code"

Die Standardwerte sind:

"1200, N, 8, 1; A, C, &1A"

Baud-Rate	1200
Parität	N (none = keine)
Wortlänge	8 Bit
Stop-Bits	1
Codetyp	A (ASCII)
Endcode	C (Carriage Return)
Textendecode	&1A (Dez 26)

- \* Jede der Optionen kann im OPEN-Kommando weggelassen werden. Der vorgegebene Wert wird dann eingesetzt.

**Beispiel:**

OPEN "...,2"

Nur die Zahl der Stop-Bits wird auf 2 gesetzt, alle anderen Optionen bleiben unverändert.

Baud-Rate: 300, 600, 1200

Setzt die Übertragungsrate. Der Computer kann mit 300, 600 oder 1200 Baud Daten senden und empfangen (1 Baud = 1 Bit/sec).

Parität: N, E, O

Setzt den Typ der Parität. N: NONE – es wird kein Paritäts-Bit benutzt. E: EVEN – setzt gerade Parität. O: ODD -- setzt ungerade Parität.

Wortlänge: 7, 8

Setzt die Zahl der Bit pro Zeichen. Es können 7 oder 8 Bit pro Zeichen gesetzt werden.

Stop-Bits: 1, 2

Setzt die Zahl der Stop-Bits.

Codetyp: A, B

Spezifiziert das zu übertragende und empfangende Code-System.

A: Spezifiziert ASCII-Codes.

B: Spezifiziert Zwischencodes.

Die Wortlänge kann für Zwischencodes nicht auf 7 Bits gesetzt werden, weil dann ERROR 1 entsteht.

Endcode: C, F, L

Setzt den Code für das Zeichen "Ende des Datensatzes". C: Setzt den Code CR (Carriage Return = Wagenrücklauf). F: Setzt den Code LF (Line Feed = Zeilenvorschub). L: Setzt den Code CR + LF.

Textendecode: &00 bis &FF (Dez 0 bis 255)

Setzt den Code für das Zeichen "Ende des Textes", um z. B. das Ende eines Programmes erkennen zu können. Kann bei den Kommandos SAVE und LOAD gefordert sein.

\* Wird das OPEN-Kommando gegeben, obwohl die Leitung bereits geöffnet war, so erhält man die Fehlermeldung ERROR 8.

Schließen Sie die Leitung mit dem CLOSE-Kommando. Die Leitung wird auch geschlossen, wenn das RUN-Kommando gegeben wird, das Programm endet, der Rechner eingeschaltet wird oder der Computer in den CAL-Modus geschaltet wird. Die gesetzten Optionen des Interface bleiben auch nach dem Schließen der Leitung erhalten.

**OPEN\$****1. OPEN\$**

Abkürzung: OP.\$, OPE.\$

Vergleiche: OPEN

**Wirkung:**

Gibt die gesetzten Optionen des seriellen Ein/Ausgabe-Interface auf der Anzeige aus.

**Anwendung:**

Die gesetzten Optionen des Interface werden als String auf der Anzeige ausgegeben.

**Beispiel:****OPEN\$ **[ENTER]******Ausgabe:****12, N, 8, 1, A, C, &1A**

**Hinweis:** Während der manuellen Ausführung wird die Anzeige der Baudrate vereinfacht. Abhängig von der Einstellung wird 3, 6 oder 12 angezeigt. (3 bedeutet 300 Baud, 6 bedeutet 600 Baud und 12 bedeutet 1200 Baud.)

**PRINT#1**

1. PRINT#1 variable, variable, variable, ...

Abkürzung: P.#1, PR.#1, PRI.#1, PRIN.#1

Vergleiche: OPEN, INPUT#1, PRINT#

**Wirkung:**

Sendet Daten über das serielle Ein-/Ausgabe-Interface.

**Anwendung:**

Das Kommando wird nur ausgeführt, wenn die Leitung zum Interface mit dem OPEN-Kommando geöffnet wurde. Sonst wird es ignoriert.

**Arten des ASCII-Codes**

\* Folgende VariablenTypen sind vorgegeben:

**Feste Variable (Konstante)**

Jeder Variablenname muß angegeben werden.

Beispiel: A, B, C\$

**Achtung:** Diese Variablen können nicht die Form A\* haben.

**Einfache Variable**

Jeder Variablenname muß angegeben werden.

Beispiel: AA; B!\$, C2

**Feld-Variable (Arrays)**

Müssen in der Form eines Feld-Namen (\*) angegeben werden.

Beispiel: B(\*), C\$(\*)

Alle Elemente eines Feldes werden in dieser Form übertragen. Einzelne Feld-Elemente können nicht direkt aus dem Feld gesendet werden.

Beispiel: 50 PRINT#1 A, AB, C\$, E(\*)

\* Der Endcode wird an das Ende einer jeden gesendeten Variablen oder eines jeden gesendeten Feld-Elements angehängt.

Beispiel:

Wenn A=12345 und B\$="ABC" ist, dann werden die Daten des Kommandos

PRINT#1A, B\$

in dieser Form gesendet (Endcode sei CR):

1.23450000E04"CR"ABC"CR"

- \* Ist der Wert einer numerischen Variablen negativ, so wird vor der Zahl ein Minuszeichen (-) gesendet.

- \* Die Elemente eines Feldes werden in dieser Form übertragen:

**Beispiel 1:**

Eindimensionales Feld B(3):  $B(\emptyset) \rightarrow B(1) \rightarrow B(2) \rightarrow B(3)$

**Beispiel 2:**

Zweidimensionales Feld (2, 3):  $C(\emptyset, \emptyset) \rightarrow C(\emptyset, 1) \rightarrow C(\emptyset, 2) \rightarrow C(\emptyset, 3) \rightarrow C(1, \emptyset) \dots$

#### Zwischencodes

- \* Die Spezifikation von Variablen ist so gut wie identisch mit ASCII mit der Ausnahme, daß feste Variable auch als A\* spezifiziert werden können. (Siehe PRINT Befehl.)
- \* Variable werden entsprechend dem internen Format gesendet, aber der Endcode wird nicht gesendet. Dem Anfang jeder Variablen wird ein Label (Gesamtzahl der gesendeten Bytes, Stringelemente, sonstige) angehängt. Wenn Variable im A\* Format spezifiziert sind, wird am Ende der CR-Code gesendet.

#### Achtung:

- \* Erweiterte Variable, wie A(27) oder darüber sowie einfache und Feld-Variablen, müssen im Programm-/Datenspeicherbereich abgelegt sein, bevor sie gesendet werden können. Im anderen Fall erhalten Sie eine Fehlermeldung.
- Sie erhalten auch eine Fehlermeldung, wenn der Typ der Variablen im Computer nicht mit dem Typ der zu sendenden übereinstimmt.
- \* Sind in den Daten Sonderzeichen, wie z.B.  $\pi$  oder das Wurzelzeichen, enthalten, werden sie in die entsprechende Kurzform, hier PI bzw. SQR, umgewandelt und dann gesendet.

## SAVE

### 1. SAVE

Abkürzung: SA., SAV.,  
Vergleiche: OPEN, LLIST

#### Wirkung:

Sendet Programme aus dem Computer über das serielle Ein-/Ausgabe-Interface.

#### Anwendung:

Wenn die Leitung zum seriellen Interface mit dem OPEN-Kommando geöffnet ist, werden Programme übertragen. Ist die Leitung geschlossen, wird das Kommando ignoriert.

- \* Das SAVE-Kommando wird nicht ausgeführt, wenn ein Passwort gesetzt wurde.
- \* Zwischen den beiden Code-Systemen bestehen die folgenden Unterschiede.

ASCII-Code: Sendet am Ende jeder Zeile einen Endcode. Sendet am Ende einer Textübertragung einen Text-Endcode.

Zwischencode: Sendet keine Endcodes und Text-Endcodes.

### 2.3.5 TEXT-FUNKTIONSKOMMANDOS

#### BASIC

##### 1. BASIC

Abkürzung: BA., BAS., BASI.

Vergleiche: TEXT

##### Wirkung:

Löscht den TEXT-Modus, nur in der Direkteingabe im PROGRAMM-Modus erlaubt. Der Wechsel aus dem TEXT-Modus in den BASIC-Modus kann durch die Tastenkombination **ESC** oder **ENTER** erfolgen. Beim ersten Aufruf des Kommandos wird der Modus automatisch auf BASIC gesetzt.

##### Anwendung:

Die Ausführung des BASIC-Kommandos löscht den TEXT-Modus und schaltet den Computer in den BASIC-Modus zurück.

Das Bereitschaftssymbol wechselt von "<" nach ">".

\* Der Wechsel aus dem TEXT- in den BASIC-Modus wandelt einen Text im Computer in ein Programm um (Interner Code).

Alle Kleinbuchstaben mit Ausnahme der in Zeichenstrings in Anführungsstrichen (" ") eingeschlossenen und der in REM-Anweisungen stehenden werden automatisch in Großbuchstaben umgewandelt.

Abkürzungen, wie z.B. "P." oder "I.", werden nicht in die entsprechenden Kommandos, hier PRINT bzw. INPUT, übersetzt. Führen Sie in diesem Fall den Cursor in die entsprechende Zeile und drücken Sie dann **ENTER**. Kommandos und Formate, die nicht im Computer vorgesehen sind, können nicht ausgeführt werden.

\* Während der Umformung wird ein Stern (\*) am rechten Ende des Bildschirms angezeigt.

\* Wurde ein Passwort gesetzt, erhalten Sie auf das Kommando BASIC die Fehlermeldung ERROR 1.

## TEXT

### 1. TEXT

Abkürzung: TE., TEX.

Vergleiche: BASIC

#### Wirkung:

Schaltet den Computer in den TEXT-Modus, nur erlaubt in Direkteingabe im PROGRAMM-Modus.

#### Anwendung:

Die TEXT-Funktion (oder -Modus) wird gebraucht, wenn ein Programm in den Computer eingegeben werden soll, das für einen Personalcomputer geschrieben wurde. Die Eingabe kann auch über das serielle Ein-/Ausgabe-Interface erfolgen.

- \* Die Ausführung des TEXT-Kommandos schaltet den Computer in den TEXT-Modus. Daten, bestehend aus einer Zahl, der Zeilennummer und die Programm-information, Befehl und Kommandos werden in den Rechner eingegeben. Dann wird die **ENTER**-Taste gedrückt, um die Daten in den Programm-/Datenspeicherbereich zu übertragen.

Natürlich können Befehle, die nicht im Computer vorgesehen sind, in den internen Code des Rechners übertragen werden.

Der Text wird so, wie er eingegeben wurde, als Folge von ASCII-Zeichen gespeichert. Er wird in der Reihenfolge der als Zeilennummern vorgesehenen Zahlen sortiert (Zeilennummer-Editierfunktion).

- \* Ein Text wird in der eingegebenen Form gespeichert.

Deshalb werden BASIC-Abkürzungen, wie z.B. "I." für "INPUT", so wie sie eingegeben werden, angezeigt und gespeichert.

- \* Ein Programm, das im internen Code des Computer gespeichert ist, wird bei Umschaltung in den TEXT-Modus in den ASCII-Code umgewandelt.
- \* Während der Programmumwandlung wird am rechten Rand ein Stern (\*) gezeigt.
- \* Das Bereitschaftssymbol des TEXT-Modus ist "<", das des BASIC-Modus ist ">".
- \* Inklusiv der Zeilennummer und dem **ENTER** darf eine Zeile im TEXT-Modus nicht länger als 80 Zeichen sein. Überzählige Zeichen werden gelöscht.

- \* Die Zahl der Bytes erhöht sich, wenn ein Programm aus dem internen Code in den ASCII-Code übertragen wird, da im internen Code ein Kommando, z.B. "PRINT", nur ein Byte, im ASCII-Code aber 6 Byte lang ist. Wird durch die Umwandlung der Speicherbereich überschritten, wird die Umwandlung rückgängig gemacht und die Fehlermeldung **ERROR 6** ausgegeben.
- \* Wurde ein Passwort gesetzt, wird auf das Kommando **TEXT** die Fehlermeldung **ERROR 1** ausgegeben.

## 2.4 FEHLERSUCHE

In diesem Kapitel sollen Sie einige Hinweise erhalten, was Sie unternehmen können, wenn Ihr SHARP PC-1450 nicht tut, was Sie von ihm erwarten. Es ist in zwei Teile gegliedert – der erste beschäftigt sich mit allgemeiner Bedienung des Gerätes, und der zweite mit der BASIC-Programmierung. Für jedes Problem schlagen wir eine Reihe von Lösungen vor. Sie sollten jede von ihnen versuchen – aber nur eine zur Zeit –, bis Sie das Problem gelöst haben.

### BEDIENUNG DES GERÄTS

#### Wenn:

Sie das Gerät einschalten, aber nichts in der Anzeige erscheint.

Die Anzeige zwar funktioniert, aber keine Reaktion auf Tastendruck erfolgt.

Sie eine Rechnung oder Antwort eingeben, aber keine Reaktion erfolgt.

Sie ein BASIC-Programm abarbeiten lassen, etwas angezeigt wird und das Programm dann anhält.

#### Dann sollten Sie:

- 1) Überprüfen, ob Sie das Gerät eingeschaltet haben (ON-Position).
- 2) Die **BRK**-Taste betätigen, um festzustellen, ob sich das Gerät automatisch abgeschaltet hatte.
- 3) Die Batterien wechseln.
- 4) Den Kontrast einstellen.
  - 1) **CLS** drücken, um die Anzeige zu löschen.
  - 2) **CA** ( **SHIFT** **CLS** ) drücken, um die Anzeige zu löschen.
  - 3) Das Gerät aus- und wieder einschalten.
  - 4) Irgendeine Taste drücken, festhalten und den ALL-RESET-Schalter drücken.
  - 5) Den ALL-RESET-Schalter ohne zusätzlichen Tastendruck betätigen.
- 1) Drücken Sie **ENTER**.
- 1) Drücken Sie **ENTER**.

Sie eine Rechnung eingeben und diese in der Form eines BASIC-Statements angezeigt wird (Doppelpunkt nach der ersten Zahl).

Sie keinerlei Reaktion auf die Tastenbetätigung erhalten.

1) Schalten Sie vom PROgramm- in den RUN-Modus um.

1) Halten Sie irgendeine Taste fest und drücken Sie den ALL-RESET-Schalter.

2) Wenn Sie dann noch immer keine Reaktion erhalten, drücken Sie den ALL-RESET-Schalter, ohne dabei eine andere Taste festzuhalten. Dabei werden jedoch alle Daten, Programme und Speicherinhalte gelöscht.

## FEHLERSUCHE IM BASIC

Wenn Sie ein neues BASIC-Programm eingeben, wird dieses in der Regel beim ersten Startversuch nicht laufen. Selbst, wenn Sie nur ein Programm abtippen, von dem Sie wissen, daß es korrekt ist, wie z.B. die in diesem Handbuch vorgestellten, dürfte Ihnen normalerweise mindestens ein Tippfehler unterlaufen. Handelt es sich um ein längeres Programm, wird es oft auch mindestens einen logischen Fehler enthalten. Es folgen einige grundsätzliche Hinweise, wie Sie solche Fehler finden und korrigieren.

Sie starten Ihr Programm und erhalten eine Fehlermeldung:

1. Schalten Sie zurück in den PROgramm-Modus und benutzen Sie die **↑** oder **↓**-Taste, um die fehlerhafte Zeile ins Display zu rufen. Der Cursor wird sich an der Stelle befinden, wo Ihr Computer verwirrt wurde.
2. Wenn Sie aus der Art, wie die Programmzeile geschrieben ist, keinen offensichtlichen Fehler entnehmen können, kann das Problem auch an den von Ihnen verwendeten Werten liegen. So wird beispielsweise CHR\$(A) eine Fehlermeldung bewirken, wenn A den Wert 1 hat, weil CHR\$(1) ein unerlaubtes Zeichen ist. Überprüfen Sie die Werte der von Ihnen verwendeten Variablen, indem Sie entweder im RUN- oder im PROgramm-Modus die einzelnen Variablennamen gefolgt von **ENTER** eingeben.

Sie starten das Programm mit RUN und erhalten keine Fehlermeldung, doch das Programm tut nicht, was Sie von ihm erwarten:

3. Überprüfen Sie das Programm Zeile für Zeile unter Verwendung von LIST und den **↓**- und **↑**-Tasten, um herauszufinden, ob Sie das Programm korrekt

eingegeben haben. Es ist erstaunlich, wieviele Fehler beim bloßen erneuten Durchsehen eines Programms gefunden werden können!

4. Versuchen Sie, jede einzelne Zeile beim Durchsehen so zu interpretieren, als wären Sie ein Computer. Nehmen Sie einfache Werte und realisieren Sie die Operationen der einzelnen Zeilen, um herauszufinden, ob Sie die gewünschten Ergebnisse erhalten.
5. Fügen Sie eines oder mehrere zusätzliche PRINT-Statements in Ihr Programm ein, um die einzelnen Werte und Tastenbelegungen zur Anzeige zu bringen. Benutzen Sie diese, um die korrekten Teile des Programms von den möglicherweise fehlerhaften zu isolieren. Diese Vorgehensweise ist auch nützlich um zu bestimmen, welche Teile des Programms schon abgearbeitet wurden. Sie können den Programmablauf auch an kritischen Stellen vorübergehend mit STOP unterbrechen, um dann einzelne Variable zu überprüfen.
6. Verwenden Sie TRON und TROFF, entweder als Befehle oder als Programmbestandteile, um den Programmablauf durch die einzelnen Zeilen hindurch verfolgen zu können. Halten Sie das Programm an kritischen Punkten an, um den Inhalt von wichtigen Variablen zu überprüfen. Dies ist zwar eine sehr langsame Methode, Fehler aufzuspüren — aber es ist manchmal die einzige.



**ANHANG**

## ANHANG A

FEHLERMELDUNGEN

Fehlernummer	Bedeutung
1	<p>Syntax-Fehler</p> <p>Der Computer kann nicht verstehen, was Sie eingegeben haben. Prüfen Sie die Eingabe auf Dinge wie Semikolon am Ende eines PRINT-Statements, falsch geschriebene Befehle und fehlerhaften Gebrauch von Zeichen.</p> <p>(z.B.: 3 * /2)</p>
2	<p>Rechenfehler</p> <p>Hier haben Sie vermutlich eines der drei folgenden Dinge getan:</p> <ol style="list-style-type: none"> <li>1. Versucht, eine zu große Zahl zu benutzen; das Rechenergebnis ist größer als 9.999999999 E 99.</li> <li>2. Versucht, durch Null zu teilen.</li> </ol> <p>(z.B.: 5 / 0)</p> <ol style="list-style-type: none"> <li>3. Versucht, eine unlogische Rechnung aus führen.</li> </ol> <p>(z.B.: LN - 30 oder ASN 1.5)</p>
3	<p>Unzulässige Funktion (DIMensionierungs-Fehler, Argument-Fehler)</p> <p>Die Feld-Variable existiert bereits. Sie haben versucht, ein Feld zu spezifizieren, ohne es zuvor zu dimensionieren.</p> <p>Der Index des Feldes übersteigt die im DIM-Statement vorgegebene Größe.</p> <p>(z.B.: DIM B(256))</p> <p>Unzulässiges Funktionsargument, d.h. Sie haben versucht, Ihren Computer etwas tun zu lassen, wozu er nicht imstande ist.</p> <p>(z.B.: WAIT 66000)</p> <ul style="list-style-type: none"> <li>• Wenn der LLIST-Befehl für Ausgabe über die serielle E/A-Schnittstelle verwendet wird, sollte der CONSOLE-Wert mit dem CONSOLE-Befehl auf 23 Zeichen pro Zeile eingestellt sein.</li> </ul>

- Wenn der CONSOLE-Wert auf 6 oder weniger Zeichen pro Zeile eingestellt ist, kann der PC-1450 für manuelles Drucken auf dem CE-140P verwendet werden.

4

#### Zu hohe Zeilennummer

Hier haben Sie vermutlich einen der folgenden Fehler gemacht:

1. Versucht, mit GOTO, GOSUB, RUN, LIST, THEN etc. eine nicht vorhandene Zeile anzusprechen.
2. Versucht, eine zu große Zeilennummer zu verwenden. Die höchstmögliche Zeilennummer ist 65279.

5

#### Verschachtelungsfehler

Das Einrichten einer Subroutine übersteigt 10 Ebenen.

Das Einrichten einer FOR-Schleife übersteigt 5 Ebenen.

Die Befehle RETURN ohne GOSUB, NEXT ohne FOR, READ ohne DATA wurden benutzt.

Fassungsvermögen des Puffers überschritten.

6

#### Speicherkapazität überschritten

Diese Fehlermeldung erhalten Sie normalerweise, wenn Sie versuchen, ein Feld zu DIMensionieren, das zu groß für die Speicherkapazität ist, oder wenn ein Programm zu lange ist.

7

#### PRINT USING Fehler

Das bedeutet, daß in einem USING-Statement eine unzulässige Format-Spezifikation enthalten ist.

8

#### E/A-Anschluß-Fehler

Dieser Fehler kann nur auftreten, wenn Sie den zusätzlichen Drucker und/oder Cassettenrecorder an den Computer angeschlossen haben.

Dieser Fehler kann auch auftreten, wenn Sie den seriellen E/A-Anschluß benutzen.

Die Meldung informiert Sie über ein Übermittlungsproblem zwischen dem Drucker und/oder Cassettenrecorder und dem Computer.

**Ausgabe 9 Fehlercode "Anderer Fehler" (L10102)**

Der Fehlercode "Anderer Fehler" ist ein Fehlercode, der von dem Computer generiert wird.

Dieser Fehlercode wird angezeigt, wenn der Computer ein Problem hat, das mit den anderen Fehlercodes nicht zu erfassen ist. Eine der häufigsten Ursachen für das Auftreten dieser Meldung ist der Versuch, Daten einer Variablen unter einem bestimmten Namen anzusprechen (z.B. A\$), während die Daten der Variablen unter dem anderen Namen (d.h. A) abgespeichert worden waren.

Ein weiterer möglicher Fehlercode ist der Fehlercode "AUSGABE".

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

**Ausgabe 10 Fehlercode "Formatfehler" (L10103)**

Der Fehlercode "Formatfehler" ist ein Fehlercode, der von dem Computer generiert wird.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

Ein Fehlercode ist eine spezielle Art von Fehlermeldung, die

die Art und Weise des Fehlers im Computer beschreibt.

## ANHANG B ASCII-CODE TABELLE

Die folgende Tabelle enthält die Umwandlungswerte für CHR\$ und ASC. Die Spalten geben das erste hexadezimale Zeichen bzw. die ersten vier Binärbits, die Zeilen das zweite hexadezimale Zeichen bzw. die letzten vier Binärbits. Oben links in jedem Kästchen ist der Dezimalcode für das jeweilige Zeichen angegeben. Das Zeichen selbst befindet sich unten rechts. Wenn kein Zeichen angegeben ist, ist dieses Zeichen nicht verfügbar. Beispielsweise ist das Zeichen "A" dezimal 65, hexadezimal 41 und binär 01000001. Das Zeichen "√" ist dezimal 252, hexadezimal FC und binär 11111100.

### Hinweis:

- \* Die Zeichen für die Zeichencodes 92 (&5C), 249 (&F9) und 250 (&FA) unterscheiden sich in der Anzeige auf dem Display vom Ausdruck mit den Druckern CE-126P und CE-140P.
- \* Das Zeichen für den Zeichencode 92 (&5C) wird auf dem Drucker CE-126P als ↴ und auf dem Drucker CE-140P als \ ausgegeben.
- \* Bei Verwendung des Druckers CE-126P darf der Code 0 (&00) keinesfalls verwendet werden.

## Erste 4 Bits

Der PC-1450 verfügt nicht über die Codes in den Schattierten Bereichen. Wenn Code-Nummern aus den schattierten Bereichen eingegeben werden, tritt ein Fehler auf.

hex binär	0	1	2	3	4	5	6	7	8	E	F
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1110	1111
0	0	16	32	48	64	80	96	112	128	224	240
	0000	NUL	SPACE	@	P	€	p	¶	„	„	„
1	1	17	33	49	65	81	97	113	129	225	241
	0001		!	1	A	Q	a	q	„	„	„
2	2	18	34	50	66	82	98	114	130	226	242
	0010		;	2	B	R	b	r	„	„	„
3	3	19	35	51	67	83	99	115	131	227	243
	0011		#	3	C	S	c	s	„	„	„
4	4	20	36	52	68	84	100	116	132	228	244
	0100		\$	4	D	T	d	t	„	„	„
Z	5	5	21	37	53	69	85	101	117	133	229
weite				%	5	E	U	e	u	„	„
4	6	6	22	38	54	70	86	102	118	134	230
	0110			&	6	F	V	f	v	♥	♥
B	7	7	23	39	55	71	87	103	119	135	231
it				7	G	W	g	w	♦	♦	♦
S	8	8	24	40	56	72	88	104	120	136	232
	1000		I	8	H	X	h	x	♣	♣	♣
A	9	9	25	41	57	73	89	105	121	137	233
	1001		)	9	I	Y	i	y	■	■	■
B	10	26	42	58	74	90	106	122	138	234	250
	1010		*	:	J	Z	j	z	□	□	□
	1011		11	27	43	59	75	91	107	123	235
					+	;	K	[	k	{	π
C	12	28	44	60	76	92	108	124	140	236	252
	1100		,	<	L	\	i	!	„	„	√
D	13	29	45	61	77	93	109	125	141	237	253
	1101		-	=	M	]	m	}	„	„	„
E	14	30	46	62	78	94	110	126	142	238	254
	1110		.	>	N	Λ	n	~	„	„	„
F	15	31	47	63	79	95	111	127	143	239	255
	1111		/	?	O	-	o	„	„	„	„

## ANHANG C

### FORMATIEREN DER DATENAUSGABE

Es ist manchmal wichtig oder nützlich, neben dem Inhalt ausgegebener Daten auch das Format zu kontrollieren. Der Computer bedient sich hierzu des Befehls **USING**. Mit Hilfe dieses Befehls können Sie spezifizieren:

- die Anzahl der Stellen
- die Position des Dezimalpunktes
- wissenschaftliche Notation
- Anzahl der Zeichen in einem String

Diese verschiedene Formate bestimmen Sie mit einer "Ausgabe Maske", die aus einer String-Konstanten oder auch einer String-Variablen bestehen kann.

1@: USING "####"

2@: M\$ = "&&&&"

3@: USING M\$

Wird der Befehl **USING** ohne Maske benutzt, werden damit alle Spezialformate aufgehoben.

4@: USING

Der Befehl **USING** kann auch in einem **PRINT**-Statement benutzt werden.

5@: PRINT USING M\$; N

Wann immer **USING** benutzt wird, bewirkt es die Kontrolle aller ausgegebenen Daten, bis ein neuer **USING**-Befehl auftritt.

### NUMERISCHE DATEN

Eine numerische **USING**-Maske darf nur benutzt werden, um numerische Werte, d.h. numerische Konstante oder numerische Variable zu kontrollieren. Wird eine String-Konstante oder -Variable angezeigt, während eine numerische **USING**-Maske wirksam ist, wird die Maske hierauf nicht angewendet werden.

Ein auszugebender Wert muß immer in den von der Maske vorgesehenen Freiraum passen. Es muß auch Raum für das Vorzeichen vorgesehen sein, selbst, wenn die Zahl immer positiv sein wird. Eine Maske, die vier Stellen zuläßt, kann also nur für dreistellige Zahlen verwendet werden.

## BESTIMMUNG NUMERISCHER MASKEN

Die gewünschte Anzahl von Stellen wird mit Hilfe des "#" -Zeichens festgelegt. Die Anzeige oder der Ausdruck beinhalten immer soviele Zeichen, wie in der Maske vorgesehen sind. Dabei erscheint die Zahl rechts in diesem Bereich, die restlichen Stellen werden durch Leerschritte aufgefüllt. Bei positiven Zahlen wird sich daher auf der linken Seite der Anzeige immer mindestens ein Leerschritt befinden. Da der Computer nicht mehr als 10 Stellen erfassen kann, sollte die größte verwendete numerische Maske maximal 11 "#" -Zeichen enthalten.

Wenn die Gesamtstelligenzahl des ganzzahligen Teils 11 Stellen übersteigt, wird dieser ganzzahlige Teil als 11 Stellen aufgefaßt.

Anmerkung: In allen Beispielen dieses Anhangs werden Sie am Anfang und am Ende eines Anzeigefeldes "!" -Zeichen finden, mit denen die Größe des Feldes anschaulich gemacht werden soll.

### EINGABE

10: USING "#####"

20: PRINT 25

30: PRINT -350

40: PRINT 1000

### ANZEIGE

(Bringen Sie den Computer in den RUN-Mode, geben Sie RUN ein und drücken Sie die **ENTER** -Taste.)

| 25 |

| -350 |

ERROR 7 in 40

Beachten Sie, daß die letzte Eingabe eine Fehlermeldung hervorruft, weil 5 Stellen (4 Zahlen und eine Stelle für das Vorzeichen) benötigt werden, die Maske aber nur 4 vorsieht.

## BESTIMMUNG DES DEZIMALPUNKTES

Das Zeichen für den Dezimalpunkt (".") kann in einer Maske enthalten sein, um die gewünschte Position des Dezimalpunktes festzulegen. Wenn die Maske mehr Stellen bereitstellt, als für den anzugebenden Wert benötigt werden, werden die auf der rechten Seite übrigbleibenden Stellen mit Nullen aufgefüllt. Enthält der anzugebende Wert mehr Stellen, als die Maske vorsieht, werden diese abgeschnitten (nicht gerundet).

### EINGABE

10: USING "#####.##"

20: PRINT 25

30: PRINT -350.5

40: PRINT 2.547

### ANZEIGE

| 25.00 |

| -350.50 |

| 2.54 |

## BESTIMMUNG DER WISSENSCHAFTLICHEN NOTATION

Das “^”-Zeichen kann in einer Maske enthalten sein, um so anzuzeigen, daß die Zahl in wissenschaftlicher Notation ausgegeben werden soll. Die Zeichen “#” und “.” werden in der Maske benutzt, um den “charakteristischen” Teil der Zahl (d.h. den Teil, der links vom “E” steht) zu formatieren. Links vom Dezimalpunkt sollten immer 2 “#”-Zeichen stehen, um genug Raum für eine ganze Zahl und das Vorzeichen zu schaffen. Der Dezimalpunkt kann – muß aber nicht – vorgegeben werden.

Rechts vom Dezimalpunkt können bis zu 9 Stellen eingerichtet werden (mit Hilfe von “#”-Zeichen). Nach dem “charakteristischen” Teil wird das Exponentenzeichen (E) angezeigt, gefolgt von der Stelle für das Vorzeichen und dem Exponenten (2stellig). Das kleinste von einer Maske vorzugebende Feld für wissenschaftliche Notation wäre demnach “##^”, damit werden Zahlen in der Form “2 E 99” dargestellt. Das größtmögliche Feld für wissenschaftliche Notation wäre “##.#####^”, das Zahlen wie “–1.234567890 E –12” darstellen kann.

### EINGABE

10: USING “###.##^”

20: PRINT 2

30: PRINT –365.278

### ANZEIGE

| 2.00E 00 |

| –3.65E 02 |

## BESTIMMUNG ALPHANUMERISCHER MASKEN

String-Konstanten und -Variable werden mit Hilfe des “&”-Zeichens ausgegeben. Jedes “&” gibt eine im vorgesehenen Feld anzuzeigende Stelle an. Der String wird auf der linken Seite dieses Feldes angezeigt. Ist der String kürzer als der dafür eingeräumte Raum, werden die rechts verbleibenden Stellen mit Leerschritten aufgefüllt. Ist der String länger als das Feld, wird er abgeschnitten:

### EINGABE

10: USING “&&&&&”

20: PRINT “ABC”

30: PRINT “ABCDEFGHI”

### ANZEIGE

| ABC |

| ABCDEF |

## GEMISCHTE MASKEN

In den meisten Anwendungsfällen wird eine USING-Maske entweder alle notwendigen Zeichen zur Formatierung eines numerischen Ausdrucks oder zur Formatierung eines Strings beinhalten. Für bestimmte Zwecke können aber auch beide gemeinsam in einer USING-Maske enthalten sein. In solchen Fällen markiert jede Umschaltung von String-formatierenden Zeichen auf Zahlen-formatierende

Zeichen (und umgekehrt) die Grenze für einen bestimmten Wert. So können mit einer Maske der Form "#####&##" zwei verschiedene Werte formatiert werden – ein numerischer Ausdruck, für den 5 Stellen vorgesehen sind, und ein alphanumerischer Ausdruck, für den 4 Stellen bereitgestellt wurden.

<u>EINGABE</u>	<u>ANZEIGE</u>
10: PRINT USING "###.##&"; 25;"CR"	25.00CR
20: PRINT -5.789;"DB"	-5.78DB

**Achtung:** Wurde ein USING-Statement einmal spezifiziert, wirkt es sich auf alle nachfolgenden Daten aus, bis es aufgehoben oder durch einen anderen USING-Befehl ersetzt wird.

#### Zur Beachtung bei Verwendung des USING-Befehls

Bei Bestimmung des Anzeigeformats mit USING müssen die folgenden Punkte beachtet werden.

1. Wenn die Anzahl der Zeichen mit einem USING-Format in PRINT ausdrück 16 Zeichen überschreitet, tritt der Fehler ERROR 7 auf.

Beispiel: 10: USING "#####.#####"

17 Zeichen oder mehr

20: PRINT A

Hinweis: Wenn die Gesamtanzahl der im ganzzahligen Teil spezifizierten Stellen einschließlich des Vorzeichens 11 überschreiten, wird der überzählige Teil ignoriert. Wenn die Gesamtanzahl der im ganzzahligen Teil spezifizierten Stellen 11 überschreiten, wird der ganzzahlige Teil als 11 Stellen angesehen.

Beispiele:

1) 10: USING "#####.#####"

Dieses Format bewirkt keinen Fehler.

2) 10: USING "#####.#####.#####"

Auch dieses Format bewirkt keinen Fehler. Ganzzahliger Teil: 11 Stellen, Dezimalpunkt: 1 Stelle, Nachkommateil: 4 Stellen

Insgesamt: 16 Stellen (weniger als 17 Stellen)

3) 10: USING "#####.#####.#####"

Dieses Format bewirkt den Fehler ERROR 7. (Gesamtanzahl der Stellen größer als 16.)

- Das USING-Format so ändern, daß es 16 oder weniger Stellen enthält.

2. Wenn die Anzahl der Stellen in einem **USING**-Format in **PRINT ausdruck, ausdruck** 8 Stellen überschreitet, entsteht ERROR 7.

Beispiel: 10: **USING "#####.##"**

9 Stellen oder mehr

20: **PRINT A, B**

- Das **USING**-Format so ändern, daß es 8 Stellen oder weniger enthält.

3. Wenn der Anzeige-Inhalt von **PRINT ausdruck; ausdruck; ausdruck; . . .** 16 Stellen überschreitet, wird der überzählige Teil nicht angezeigt.

4. Die Anzeige von Werten, die in hexadezimale Darstellung umgewandelt wurden, wenn das Fließkommasystem (Exponentialanzeige) mit einer **USING**-Anweisung gewählt wurde, bewirkt einen Fehler. (ERROR 7, Formatfehler)

Unter dem Titel „**ANHANG D**“ befindet sich ein Abschnitt mit dem Titel:

## **BEWERTUNG VON AUSDRÜCKEN UND OPERATOREN-VORRANG**

Wenn in den Computer ein komplexer Ausdruck eingegeben wird, bewertet er Teile dieses Ausdrucks in einer Reihenfolge, die durch die Vorrangstellung der einzelnen Teile bestimmt wird. Geben Sie den Ausdruck

$$100/5+45$$

ein, entweder als Rechenoperation oder als Teil eines Programms, so weiß der Computer nicht, ob Sie meinen:

$$\frac{100}{5} + 45 = 2$$

$$\text{oder } \frac{100}{5+45} = 65$$

Da der Computer eine Möglichkeit haben muß, zwischen diesen beiden Operationen zu entscheiden, bedient er sich seiner Regeln des Operatoren-Vorrangs. Da die Division eine höhere "Priorität" hat als die Addition (siehe unten), wird er entscheiden, daß zuerst die Division durchgeführt wird und anschließend die Addition, d.h. die zweite Möglichkeit wird ausgeführt und als Ergebnis 65 ausgegeben.

### **VORRANG VON OPERATOREN**

Im BASIC-Modus werden Operatoren gemäß den folgenden Prioritätsregeln verarbeitet (angefangen mit der höchsten Priorität):

1. Klammern
2. Variable und Pseudovariable
3. Funktionen
4. Exponentiation (^)
5. Vorzeichen (+, -)
6. Multiplikation und Division (\*, /)
7. Addition und Subtraktion (+, -)
8. Verhältnis-Operatoren (<, <=, =, >, >=, >)
9. Logische Operatoren (AND, OR)

Treten in einem Ausdruck zwei oder mehr Operatoren derselben Prioritätsstufe auf, wird der Ausdruck von links nach rechts verarbeitet. (Exponentiation wird von rechts nach links verarbeitet!) Beachten Sie, daß bei einer Operation A+B-C das Ergebnis dasselbe ist, ob Sie nun die Addition oder die Subtraktion zuerst ausführen.

Enthält ein Ausdruck ineinanderliegende Klammern, so wird die innerste Klammer zuerst bearbeitet, die äußerste zuletzt.

Für die Prioritätsstufen 3 und 4 gilt, daß die letzte Eingabe die höchste Priorität hat.

z. B.:

$$-2^4 \longrightarrow -(2^4)$$

$$3^{-2} \longrightarrow 3^{-2}$$

## BEISPIEL FÜR EINE BEWERTUNGSFOLGE

Wir gehen aus von dem Ausdruck:

$$((3+5-2) * 6+2)/10^{\log} 100$$

Der Computer verarbeitet nun zuerst die innersten Klammern. Da “+” und “-“ auf derselben Stufe stehen, wird von links nach rechts gerechnet, also die Addition zuerst ausgeführt.

$$((8-2) * 6+2)/10^{\log} 100$$

Dann wird die Subtraktion durchgeführt:

$$((6) * 6+2)/10^{\log} 100$$

Oder:

$$(6 * 6+2)/10^{\log} 100$$

In der nächsten Klammer wird zuerst die Multiplikation durchgeführt:

$$(36+2)/10^{\log} 100$$

Dann folgt die Addition:

$$(38)/10^{\log} 100$$

Oder:

$$38/10^{\log} 100$$

Nachdem nun die Klammern aufgelöst sind, hat die LOG-Funktion höchste Priorität.

$$38/10^2$$

Als nächstes folgt die Exponentiation:

$$38/100$$

Und zuletzt wird die Division ausgeführt:

$$0.38$$

Dies ist der Wert des Ausdrucks.

**ANHANG E TASTENFUNKTIONEN****ON  
BRK**

(ON) wird benutzt, um den Computer anzuschalten, wenn er sich automatisch abgeschaltet hatte.

**(BREAK)**

Das Betätigen dieser Taste während eines Programmablaufs bewirkt eine Unterbrechung der Programmausführung.

Bei manuellen Operationen, Ein-/Ausgabe-Befehlen wie BEEP, CLOAD etc. wird mit Betätigung dieser Taste die Befehlausführung unterbrochen.

**SHIFT**

Die gelbe Taste mit der Aufschrift "SHIFT" muß benutzt werden, um Doppelfunktionen anzusprechen (die jeweils in brauner Schrift über den entsprechenden Tasten stehenden Zeichen). Z.B. wird mit SHIFT und ?/U das "?" angesprochen.

**CLS**

Wird benutzt, um Eingabe und Anzeige zu löschen, weiterhin hebt diese Taste eventuelle Blockaden durch Fehler auf.

**SHIFT CA**

Löscht nicht nur den Anzeigehalt, sondern initialisiert darüber hinaus den Computer, d. h.

- hebt den WAIT-Timer auf;
- löscht das Anzeigeformat (USING-Format);
- hebt den TRON-Befehl auf (TROFF);
- hebt PRINT=LPRINT auf;
- hebt Blockade durch Fehler auf.

**0 bis 9** Zifferntasten**.** Dezimalpunkt

- wird benutzt, um eine Abkürzung eines Befehls, eines Kommandos oder einer Funktion einzugeben;
- gibt in der Bestimmung eines USING-Formats die Stellung des Dezimalpunktes an.

**E**

Wird benutzt, um in wissenschaftlicher Notation den Exponenten zu bestimmen (mit der Buchstabetaste "E").

**/**

Divisionszeichen.

**\***

Multiplikationszeichen

- wird benutzt, um eine Feld-Variable in INPUT#, PRINT# etc. zu bestimmen.

<b>[+]</b>	Additionszeichen
<b>-</b>	Subtraktionszeichen
<b>[SHIFT] [^]</b>	Wird benutzt, um Zahlen zu potenzieren und um das Exponenten-Anzeige-System für numerische Daten in USING-Anweisungen zu spezifizieren.
<b>[SHIFT] [&lt;]</b>	Werden benutzt, wenn logische Operatoren in einer IF-Sequenz eingegeben werden sollen.
<b>[DEF]</b>	Wenn eines der folgenden 18 Zeichen (A, S, D, F, G, H, J, K, L, =, Z, X, C, V, B, N, M, SPaCe) nach Betätigen der <b>[DEF]</b> -Taste gedrückt wird, wird das Programm mit dem entsprechenden Kennbuchstaben gestartet.
<b>[A] bis [Z]</b>	Buchstabentasten  Diese Tasten sind Ihnen wahrscheinlich von einer gewöhnlichen Schreibmaschine her vertraut.  Auf einfachen Tastendruck erscheinen Großbuchstaben in der Anzeige. In den Kleinschriftmodus schalten Sie um, indem Sie <b>[SML]</b> drücken.
<b>[SPC]</b>	Wird beim Eingeben von Zeichen oder Programmen benutzt, um einen Leerschritt zu produzieren.
<b>[=]</b>	– In Zuweisungs-Statements: weist der links vom “=” stehenden Variablen die rechts stehenden Zeichen (Buchstabe oder Zahl) zu.  – Wird weiterhin benutzt, wenn in einer IF-Sequenz logische Operatoren eingegeben werden.
<b>[SHIFT] !”#\$</b>	Zum Ansprechen der jeweiligen Zeichen:  % & @ ” – zum Ansprechen und Löschen von Zeichen; – zum Bestimmen von Kennbuchstaben; # – im USING-Statement wird hiermit die Anweisung für numerische Inhalte gegeben; \$ – wird zur Zuweisung von String-Variablen benutzt; & – im USING-Statement wird hiermit die Anweisung für Buchstabeninhalte gegeben; & – weist hexadezimale Zahlen aus; @ – wird für den Reservespeicher benutzt, wenn die Reservetaste als Programmertaste eingesetzt ist; !% – als Zeichenstring innerhalb ” ” zu verwenden.

- [SHIFT] [?] Zur Eingabe von CLOAD?**
- [: Zur Trennung von zwei oder mehr Befehlen in einer Programmzeile.]
  - [?] Bezeichnet eine Pause zwischen zwei Gleichungen, zwischen Variablen oder Kommentaren.
  - [;]
    - Zur Ausführung von Multi-Display (Anzeige von zwei oder mehr Werten / Inhalten zur gleichen Zeit)
    - Schafft eine Pause zwischen Instruktion und Variable.
- [C] [,] Zur Eingabe von Klammern.**
- [,] Bewegt den Cursor nach rechts (auf einmaligen Tastendruck Bewegung um eine Stelle; wird die Taste festgehalten, Daueraktion).
  - Zur Ausführung von Playback-Anweisungen.
  - Löscht bei manueller Rechnung Fehlermeldungen.
  - [**] Bewegt den Cursor nach links (auf einmaligen Tastendruck Bewegung um eine Stelle; wird die Taste festgehalten, Daueraktion).**
  - Sonst wie unter der [**] -Taste beschrieben.**
- [SHIFT] [INS]** Fügt einen Leerschritt ein, dabei erscheint das Zeichen "□". Dieser Leerschritt wird vor das Zeichen gesetzt, auf dem der Cursor steht.
- [SHIFT] [DEL]** Löscht den Inhalt der Stelle, auf der der Cursor steht.
- [SHIFT] [TC]** Zur Eingabe von  $\pi$  ( $\pi$ )
- [SHIFT] [F]** Zur Eingabe der Quadratwurzel
- [SHIFT] [INPUT]** BASIC-Befehlstasten; Durch Drücken von **[SHIFT]** und einer Buchstabetaste (einschließlich Komma, Leerstelle (SPC) und **[ENTER]**) wird der über der Taste angegebene Befehl in den Computer eingegeben.
- [CAL]** Zur Einstellung des CAL-Modus.
- [BASIC]** Zur Einstellung des RUN-Modus, wenn der CAL-Modus eingestellt ist.
- Zur Einstellung des PRO-Modus, wenn der RUN-Modus eingestellt ist.
- Mit jedem Drücken der Taste **[BASIC]** wird abwechselnd RUN und PRO eingestellt.
- [SHIFT] [DRG]** Zur Einstellung der Winkeleinheit.

**[hyp]** Zur Eingabe einer hyperbolischen Funktion.

**[SHIFT]** **[archyp]** Zur Eingabe einer inversen hyperbolischen Funktion.

**[sin]**  $\sim$  **[x<sup>2</sup>]**  
**[SHIFT]** **[sin<sup>-1</sup>]** Zur Eingabe der angegebenen Funktionen.

**[SHIFT]** **[n!]**

- [ENTER]**
- Zur Eingabe einer Programmzeile in den Computer.
  - Wird beim Schreiben von Programmen benötigt.
  - Bedingt manuelle Rechnung oder direkte Ausführung eines Befehls durch den Computer.
  - Zum Neustart eines Programms, das zeitweise durch einen INPUT- oder PRINT-Befehl unterbrochen wurde.

**[SHIFT]** **[P++NP]** Stellt die Verbindung zum Drucker her bzw. unterbindet sie (sofern ein zusätzlicher Drucker an den Computer angeschlossen ist).

- [SML]**
- Zum Anwählen und Aufheben des Kleinschriftmodus (schaltet die Anzeige SML an bzw. aus).
  - Die SML-Anzeige erscheint, wenn **[SML]** gedrückt wird. Wenn Sie nun die Tasten **[A]**, **[B]** und **[C]** betätigen, werden a, b und c in der Anzeige ausgegeben. Durch erneutes Drücken von **[SML]** schalten Sie den Kleinschriftmodus wieder aus, die Anzeige SML verschwindet und es werden wieder Großbuchstaben in der Anzeige ausgegeben.

Die **↑** - und **↓** -Tasten haben die folgenden Funktionen, je nach angewähltem Modus und Status des Computers:

Modus	Status	<b>↓</b>	<b>↑</b>
RUN	Programmdurchführung	um die vorherige Programmzeile anzuzeigen.	um die nächste Programmzeile anzuzeigen.
	Programm vorübergehend unterbrochen;	um die nächste Zeile durchführen zu lassen.	Diese Taste festhalten, um bereits abgearbeitete Zeilen zur Anzeige zu bringen.
	INPUT-, PRINT- Statement wird ausgeführt		
	Unterbrechung		
	Fehlermeldung während der Programmausführung		um die fehlerhafte Zeile zur Anzeige zu bringen
	TRON Status	zur Fehlersuche	Diese Taste festhalten, um bereits abgearbeitete Zeilen zur Anzeige zu bringen.

Wenn der Modus von RUN nach PRO geändert wird, wird dies nicht angezeigt!

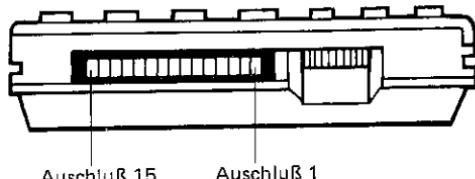
PRO	Programm vorübergehend unterbrochen	um die unterbrochene Zeile anzuzeigen	wie links
	Fehler	um die fehlerhafte Zeile anzuzeigen.	wie links
	Anderer Status	um die erste Zeile anzuzeigen.	um die letzte Zeile anzuzeigen.
(Wenn die Programmzeile angezeigt wird:)		um die nächste Programmzeile anzuzeigen.	um die vorherige Programmzeile anzuzeigen.

- In der Anzeige ist die **ENTER**-Taste das gleiche wie ein Leerschritt.
- Wenn 12 Minuten lang keine Taste betätigt wird, schaltet sich der Computer automatisch ab.

## ANHANG F

## SIGNAL DES E/A-ANSCHLUSSES

Der Computer ist mit einem 15poligen Anschluß für ein E/A-Terminal ausgestattet. Die Anschlüsse und ihre Symbole werden anschließend beschrieben.



Anschluß	Name	Signal	E/A	Funktion
1	Frame Ground	FG		Massenanschluß für Wartung
2	Transmit Data	SD	o	DC Ausgabesignal
3	Receive Data	RD	i	DC Eingabesignal
4	Request Data	RS	o	ON: Sendesignal
5	Clear To Send	CS	i	ON: Übertragung
7	Signal Ground	SG	—	Masse für alle Signale
8	Data Carrier Detect	CD	i	ON: Trägersignal erhalten
10		VC		Energieversorgung
11	Receive Ready	RR	o	ON: Empfang möglich
12	Peripheral acknowledge	PAK	i	Siehe Hinweis 3
13		VC		Energieversorgung
14	Data Terminal Ready	ER	o	ON: Terminal bereit
15	Peripheral request	PRQ	o	Siehe Hinweis 3

## Hinweis:

- (1) Der VC-Spannungspegel ist HIGH (hoch), der SG-Spannungspegel LOW (niedrig).
- (2) Der PC-1450 verwendet CMOS-Bauteile. Die Anwendung von Spannungen, die den zulässigen Bereich zwischen SG und VC überschreiten, kann den Computer beschädigen.
- (3) Dieses Signal bestätigt, ob das angeschlossene Gerät der Drucker CE-140P ist oder nicht. Dieses Signal wird nicht für die Kommunikation mit anderen Geräten verwendet.

**ANHANG G****TECHNISCHE DATEN**

Modell:	<b>PC-1450 Taschencomputer</b>
Prozessor:	8 Bit CMOS CPU
Programmiersprache:	BASIC
Speicherkapazität:	System-ROM 40 K Bytes
RAM:	ca. 1 kByte
System:	ca. 1 kByte
Anwender:	Fester Speicherbereich 208 Bytes (A-Z, A\$-Z\$) Programm-/Datenbereich 3070 Bytes (bei Verwendung der CE-211M)
Stack:	Unterroutine: 10 Stacks Funktion: 16 Stacks FOR-NEXT: 5 Stacks Daten: 8 Stacks
Operatoren:	Addition, Subtraktion, Multiplikation, Division, trigonometrische und inverstrigonometrische Funktionen, logarithmische und Exponentialfunktionen, Winkelumkehrfunktionen, Quadrat und Quadratwurzel, Vorzeichen, Absolut, Integral, Verhältnis-Operatoren, logische Operatoren.
Rechengenauigkeit:	10 Stellen (Mantisse) + 2 Stellen (Exponent)
Editiermöglichkeit:	Cursor rechts und links, Zeile auf- und abwärts, Zeicheneingabe, Zeichenlöschung
Specherschutz:	CMOS, batteriegestützt
Serieller E/A-Anschluß und seine Möglichkeiten:	
Standards:	Start-Stop-Übertragungssystem (asynchron). Nur Halbduplex.
Baud-Raten:	300, 600, 1200 Baud
Datenbits:	7 oder 8 Bits
Paritätsbits:	Gerade, ungerade oder keine Parität
Stopbit:	1 oder 2 Bits

Anschlüsse:	15poliger Anschluß (für externe Geräte)
Ausgabesignal:	C-MOS Niveau (4–6 Volt)
Interface-Signale:	Eingaben: RS, CS, CD, PAK Ausgaben: SD, RS, RR, ER, PRO Andere: SG, FG, VC
Anzeige:	16-Stellen-Flüssigkristallanzeige mit 5 x 7-Punkt-Zeichen
Tastatur:	78 Tasten, alphabetisch, numerisch, Sonderzeichen, Funktionen, numerische Schablone, definierbare Tastenfunktionen
Stromversorgung:	6,0 V DC-Lithium-Zellen Typ: CR-2032 x 2
Stromverbrauch:	6,0 V DC ca. 0,03 W Ca. 120 Stunden bei Dauerbetrieb bei einer Betriebstemperatur von 20 Grad Celsius. Diese Betriebszeit kann leicht schwanken, in Abhängigkeit von Arbeitsmethoden etc.
	* Bei einstündigem Betrieb pro Tag hält die Batterie ca. 2,5 Monate. Dies gilt, wenn die Betriebsstunde aus 10 Minuten Rechnung und 50 Minuten Anzeige besteht.
Betriebstemperatur:	0 Grad Celsius bis 40 Grad Celsius
Abmessungen:	182 (B) x 72 (T) x 16 (H) mm
Gewicht:	ca. 175 g (mit Batterien und RAM-Karte)
Zubehör:	Deckel, 2 eingebaute Lithium-Zellen, Tastaturschablone, Handbuch und RAM-Karte (CE-211M)
Zusatzeräte:	Einsteckkarte – 8 K Bytes RAM (CE-201M) – 16 K Bytes RAM (CE-202M) Cassettenrecorder (CE-152) Drucker/Cassetten-Interface (CE-126P) Drucker (CE-140P) etc.

**ANHANG H****DIE BENUTZUNG VON PROGRAMMEN, DIE FÜR  
ANDERE PC-MODELLE ENTWICKELT WURDEN**

Programme, die für die folgenden Computer geschrieben wurden, können mit geringfügigen Veränderungen auf dem PC-1450 verwendet werden.

PC-1210 Serie: PC-1210/11

PC-1245 Serie: PC-1245/46

PC-1250 Serie: PC-1250/51/55

PC-1260 Serie: PC-1260/61

PC-1350 Serie: PC-1350

PC-1401 Serie: PC-1401/02

PC-2500 Serie: PC-2500

Obwohl sich die Funktionen der obigen Modelle geringfügig unterscheiden, können die mit diesen Modellen erstellten Programme auf dem PC-1450 verwendet werden, wenn die nachstehend beschriebenen Modifikationen durchgeführt werden.

Hinweis 1: Der PC-1450 kann Programme von Cassette lesen, die mit Computern der Serien PC-1210, PC-1245 und PC-1250 auf Cassette geschrieben wurden, aber umgekehrt können Programme, die mit dem PC-1450 erstellt und auf Cassette geschrieben wurden, nicht mit den Computern der drei obengenannten Serien gelesen werden.

Hinweis 2: Programm-Cassetten für die Serien PC-1245 und PC-1210, auf die mehrere Programme mit dem MERGE-Befehl aufgenommen wurden, können nicht mit dem PC-1450 verwendet werden. Vielmehr müssen die Programme einzeln mit dem MERGE-Befehl in den PC-1450 geladen werden.

Hinweis 3: Programme, die mit der Serie PC-1250 erstellt werden und die Befehle POKE oder CALL enthalten, können nicht auf dem PC-1450 ausgeführt werden. Wird die Ausführung auf dem PC-1450 versucht, kann es vorkommen, daß der Computer keine Eingabe von den Tasten annimmt.

**MODIFIKATIONEN VON PROGRAMMEN DER PC-1245 SERIE (PC-1245,  
PC-1246)**

Wenn Sie die Programme, die für die PC-1245 Serie entwickelt wurden, auf dem PC-1450 benutzen wollen, müssen Sie folgende Änderungen vornehmen:

**1. Multiplikation ohne das " \* "-Zeichen**

Bei der PC-1245 Serie kann der Operator für die Multiplikation ( \* ) fortgelassen werden (z.B. AB anstatt A \* B, DC statt D \* C). Dies ist auf dem PC-1450 nicht

möglich, da dieser zwei aufeinanderfolgende Zeichen als einfache Variable behandelt. Benutzen Sie daher die Schreibweise rechts im Beispiel:

(z.B.)  $A = \text{SIN } BC \rightarrow A = \text{SIN } (B * C)$

## 2. Definition von Index-Variablen mit dem DIM-Befehl

Wenn auf den Computern der PC-1245 Serie der Befehl **DIM A(30)** ausgeführt wird, werden Speicherplätze von A(27) bis A(30) als Erweiterung eines Definitionsbereiches einer festen Variablen reserviert.

Auf dem PC-1450 wird jedoch bei der Ausführung von **DIM A(30)** ein getrennter Speicherbereich für die Feld-Variablen A(0) bis A(30) und das Feld mit dem Namen A reserviert.

Wenn Sie Index-Variable als Erweiterung von Festvariablen definieren wollen, benutzen Sie bitte die rechts im Beispiel angegebene Schreibweise:

(z. B.) **DIM A(30) —> A(30) = 0**

## 3. Datenein-/ausgabefehl für Dateien auf Cassette:

Bei der PC-1245 Serie wird mit dem Befehl **PRINT #C** der Inhalt der Variablen C und aller folgenden abgespeichert.

Beim PC-1450 jedoch wird mit diesem Befehl lediglich der Inhalt der Variablen C auf Band gespeichert. Um den Inhalt einer bestimmten Variablen und aller ihr folgenden auf Band abzuspeichern, benutzen Sie bitte die Schreibweise rechts im Beispiel:

(z. B.) **PRINT #A —> PRINT # A\***  
**INPUT #C —> INPUT #C\***

## 4. Wert einer Schleifenvariablen nach Beendigung einer FOR-NEXT-Schleife

Der Wert einer Schleifenvariablen, den Sie nach Ausführung einer FOR-NEXT-Schleife erhalten, unterscheidet sich beim PC-1450 von dem bei der PC-1245 Serie. Wird der Wert einer Schleifenvariablen in einem Bedingungsausdruck benutzt, müssen Sie ihn beim PC-1450 gegenüber der PC-1245 Serie um 1 erhöhen.

(z. B.) **10 FOR I=0 TO 10**  
**50 NEXT I**  
**60 IF I=10 THEN 100**

Ändern Sie den Wert in Zeile 60 folgendermaßen:

**60 IF I=11 THEN 110**

(Auf dem PC-1450 müssen Sie den Wert einer Schleifenvariablen um eins erhöhen. Die Anzahl der Schleifendurchläufe bleibt jedoch gleich.)

5. Umdefinition von '=' ist nicht möglich. Daher kann die Gleichheitstaste nicht definiert werden. Die Gleichheitstaste '=' funktioniert beim PC-1450 nicht als definable Key. Daher muß in Programmen, in denen die Gleichheitstaste definiert ist, eine andere Taste verwendet werden.

(z.B.) 100 "=";... 100 "N";... (Von der Tastatur wird das Zeichen nicht eingelesen)

6. Exponentialzeichen "E" ist nicht definiert. Daher kann es nicht benutzt werden. Der PC-1450 benützt den Großbuchstaben "E" für die Darstellung der Exponentiation. Die folgenden Änderungen werden dadurch erforderlich:

A = 1.234 IE 5  $\longrightarrow$  A = 1.234E5

B = IE 6  $\longrightarrow$  B = 1E6

Wird ein Programm der PC-1245 Serie von Band in den PC-1450 eingelesen, wird diese Veränderung automatisch vom PC-1450 durchgeführt.)

7. Der Zeichen-Code der PC-1245 Serie unterscheidet sich geringfügig von dem des PC-1450. Wenn die folgenden Codes durch die CHR\$-Funktion zugewiesen werden, ändern Sie die Codes:

ASCII	PC-1245	PC-1450
39 (&27)	[	,
91 (&5B)	√	[
92 (&5C)	¥	\
93 (&5D)	π	]
96 (&60)	IE	,
250 (&FA)	— (Error)	[
251 (&FB)	— (Error)	π
252 (&FC)	— (Error)	√

Anmerkung: Wie oben gezeigt, verfügt der PC-1450 nicht über das Zeichen IE.

## ZUSÄTZLICHE MODIFIKATIONEN

1. Die Modelle der PC-1245 Serie verfügen über eine Programmzeilenspanne von 1 – 999. Der PC-1450 hat eine erweiterte Spanne von 1 – 65279. Aus diesem Grund beansprucht die Zeilennummer 3 Bytes im RAM (bei der PC-1245 Serie nur 2 Bytes.). Die Änderung wird automatisch ausgeführt, wenn Sie Programme von Cassette in den Computer einladen. Es besteht die Möglichkeit eines Speicherüberlaufs, wenn Sie zu lange Programme laden oder ausführen lassen (ERROR 6). Wenn eine Zeile etwa 80 Bytes umfaßt, kann durch diese Veränderung das Zeilenende gelöscht werden.

2. Wenn Sie ein Programm der PC-1245 Serie in den PC-1450 einladen, wird aufgrund der Änderung in der Zeilennummerierung (von 2 auf 3 Bytes) die BUSY-Anzeige noch für 1 oder 2 Sekunden aufleuchten, nachdem das Band zu laufen aufgehört hat. Während dieser Zeit wird das “\*”-Zeichen in der rechten unteren Ecke des Zeilendisplays sichtbar sein.

### MODIFIKATIONEN FÜR PROGRAMME DER PC-1250 SERIE

Zur Verwendung von Programmen, die mit der PC-1250 Serie erstellt wurden, auf dem PC-1450, sind zusätzlich zu den für Programme der PC-1245 Serie beschriebenen Modifikationen die folgenden Modifikationen erforderlich.

Die Computer der PC-1250 Serie verfügen über eine Anzeige mit 24 Zeichen, während der PC-1450 eine 16-stellige Anzeige hat. Daher bewirken Anzeigebefehle wie PRINT, PAUSE, INPUT usw. einen Fehler, die Nichtanzeige von Teilen oder falsche Anzeigen. Daher müssen die folgenden Modifikationen durchgeführt werden.

- (1) Bei der Anzeige mit PRINT tritt ERROR 7 auf, wenn die Gesamtzahl der mit dem USING-Befehl spezifizierten Stellen 16 überschreitet. (Wenn für den ganzzahligen Teil mehr als 11 Stellen spezifiziert wurden, wird der Teil nach der 11. Stelle ignoriert.)

Überschreitet 17 Stellen

```
10 USING "#####.#####"  
20 PRINT A
```

- \* Schreiben Sie die USING-Zeile neu, so daß 16 Stellen oder weniger spezifiziert sind.

- (2) Wenn der mit USING spezifizierte ganzzahlige Teil 8 Stellen (einschließlich Vorzeichen und Dezimalpunkt) überschreitet, tritt während der Anzeige mit PRINT der ERROR 7 auf.

Überschreitet 9 Stellen

```
10 USING "#####.##"  
20 PRINT A, B
```

- \* Schreiben Sie die USING-Zeile neu, so daß 8 Stellen oder weniger spezifiziert sind.

- (3) Wenn eine Meldung (z.B. eine Eingabeaufforderung), die mit INPUT “ . . . ”; A angezeigt wird, 15 Stellen überschreitet, wird der Teil der Meldung, der die 15. Stelle überschreitet, nicht angezeigt.

- \* Verkürzen Sie die Meldung.

- (4) Zusätzlich zu den oben aufgeführten Fällen wird der Teil, der 16 Stellen überschreitet, in den folgenden Fällen nicht angezeigt.

PRINT {Ausdruck  
"Zeichen"} ; {Ausdruck  
Zeichenvariable} ; {Ausdruck  
"Zeichen"} ; ... ; {Ausdruck  
Zeichenvariable}

Bei solchen Ausdrücken überschreitet der Anzeigehinheit 16 Stellen.

PRINT {Ausdruck  
"Zeichen"} ; {Ausdruck  
Zeichenvariable} ; {Ausdruck  
"Zeichen"} ; ... ; {Ausdruck  
Zeichenvariable}

In diesem Format überschreitet der rechte oder linke Teil 8 Stellen.

INPUT "...", A

Diese Meldung überschreitet 16 Stellen.

\* In den obigen Fällen muß die Anzeige so verändert werden, daß sie nicht 16 Stellen überschreitet.

## ÄNDERUNGEN BEZÜGLICH DER PC-1210 SERIE

Um Programm der PC-1210 Serie auf dem PC-1450 zu benutzen, müssen diese in gleicher Weise geändert werden, wie Programme der PC-1245 Serie (außer die Punkte 2. und 7.). Zusätzlich sind folgende Änderungen notwendig:

### 1. IF-Statement

Beispiel:

Die folgende Programmzeile eines Programms der PC-1210 Serie

50 IF A > L PRINT "A"

wird vom PC-1450 interpretiert als

50 IF A > LPRINT "A"

und erzeugt eine Fehlermeldung.

Der Fehler taucht auf, weil ein Befehl (LPRINT) nicht in der PC-1210 Serie verwendet wird, wohl aber beim PC-1450.

Um dieses Problem zu lösen, sollten Sie in das IF-Statement einen THEN-Befehl einfügen:

50 IF A > L THEN PRINT "A"

## 2. USING-Format

Der USING-Befehl unterscheidet sich bei der PC-1210 Serie und dem PC-1450 in folgender Weise:

Beispiel:

```
10 A = -123.456
20 PAUSE USING "####.##";A
30 PAUSE A, USING "#####";A
```

Bei der Ausführung dieses Programms wird folgendes angezeigt:

* PC-1210		-123.45	
		-123	-123
* PC-1450		-123.45	
		-123.45	-123

In der PC-1210 Serie wird die Anzeige auf der linken Seite im gleichen Format angezeigt wie auf der rechten Seite. Beim PC-1450 gestaltet sich die Anzeige entsprechend dem vorangegangenen Format-Befehl.

Dies bezieht sich nicht nur auf den PAUSE-Befehl, sondern gilt auch für PRINT- und LPRINT-Befehle.

## 3. Auslassen von ")"

In der PC-1210 Serie kann das ")" weggelassen werden, sofern es direkt vor **ENTER** oder ":" vorkommt. Beim PC-1450 kann das ")"-Zeichen nicht ausgelassen werden.

Deshalb müssen Sie es einfügen, wenn Sie Programme übernehmen wollen.

## 4. PRINT-Befehl

Der PC-1450 hat einen PRINT-Befehl für die Anzeige und einen LPRINT-Befehl für die Ausgabe auf dem Drucker. Alle PRINT-Befehle können umgeändert werden in LPRINT-Befehle durch ein PRINT=LPRINT-Statement.

Die PC-1210 Serie hat keinen LPRINT-Befehl. Um bei Übernahme von Programmen auf dem PC-1450 Ausgaben auf dem Drucker zu erhalten, müssen Sie ein PRINT=LPRINT-Statement in das Programm einfügen.

## 5. Variablen

Wird ein Programm mit RUN auf der PC-1210 Serie gestartet, werden alle Variablen beibehalten. Beim PC-1450 werden alle Variablen von A(27) an gelöscht.

Deshalb ist es nötig, falls die Variablen beibehalten werden sollen, das Programm auf dem PC-1450 mittels GOTO oder der [DEF]-Taste zu starten.

## MODIFIKATIONEN FÜR DIE PC-1260 SERIE

### (1) Zeichencode-Modifikation

Der Zeichencode 96 (&60) ist in der PC-1260 Serie ein Leerzeichen, aber im PC-1450 ein Hochkomma (''). Wenn daher in einem Programm der CHR\$-Befehl mit dem Zeichencode 96 zur Spezifizierung einer Leerstelle vorkommt, muß dieser Zeichencode in 32 (&20) verändert werden.

### (2) Befehle CLS, CURSOR

Der PC-1450 verfügt nicht über die Befehle CLS und CURSOR. Daher müssen diese Befehle in Programmen, die auf dem PC-1450 laufen sollen, gelöscht und verändert werden.

## MODIFIKATIONEN FÜR DIE PC-1401 SERIE

Programme, die auf Computern der Serie PC-1401 erstellt wurden, können ohne Veränderungen auf dem PC-1450 laufen.

## MODIFIKATIONEN FÜR DIE SERIEN PC-1350 UND PC-2500

Der PC-1450 verfügt nicht über die folgenden Befehle, so daß diese verändert werden müssen, wenn sie in einem Programm vorkommen.

CLS, CURSOR, MEM\$, GCURSOR, GPRINT, LINE, POINT, PRESET, PSET, (TEST)

## PROGRAMMBEISPIELE

Beim Durcharbeiten der Bedienungsanleitung bis zu dieser Seite haben Sie viele Programmbefehle kennengelernt. Zum Programmieren in BASIC sind jedoch theoretische Kenntnisse alleine nicht ausreichend, vielmehr ist praktisches Üben erforderlich.

Auf die gleiche Weise, wie beim Lenken eines Fahrzeugs und Schwingen eines Tennisschlägers Ihre Fähigkeiten besser werden, macht auch beim Programmieren Übung den Meister.

Neben der Erstellung von eigenen Programmen ist es auch sehr wichtig, Programme anderer durchzuarbeiten, um Programmiertechniken kennen und verstehen zu lernen. Daher bringen wir auf den folgenden Seiten einige Programmbeispiele, die Ihnen hoffentlich beim Erlernen von BASIC hilfreich sein werden.

(Die SHARP CORPORATION und/oder ihre Tochtergesellschaften übernehmen keine Haftung für finanzielle oder sonstige Verluste oder Schäden, die aus der Verwendung der folgenden Programmbeispiele entstehen. Bei Verwendung dieser Programmbeispiele obliegt es dem Benutzer, die Programme auf Fehlerfreiheit zu prüfen und den jeweiligen Verwendungszwecken anzupassen.)

# INHALT

Programmtitel	Seite
● Unwandlung zwischen orthogonalen Koordinaten und Polarkoordinaten.	245
P ● Exponentielle Regression und Plotten	249
P ● Pareto-Diagramm	253
● Berechnung des Flächeninhalts eines Dreiecks.	258
● Anschmiegung eines Kreises an zwei Kreise.	262
P ● Frequenzgang-Kurve	265
P ● Kreisgrafik	271
● Datenübertragung	278
P ● 3-dimensionale Grafik	282
● Zahlenspiel	285
* Am Ende jeder Programmlistung ist der erforderliche Speicherplatz als Anzahl Bytes angegeben.	

Programme, die mit P gekennzeichnet sind, erfordern den Plotter/Drucker CE-515P oder CE-516P.

1) Verwendung des CE-515P:

Vor der Verwendung alle DIP-Schalter auf "aus" stellen.

2) Verwendung des CE-516P:

Die DIP-Schalter Nr. 1 bis Nr. 5 auf "aus" und die restlichen auf "ein" stellen.

Bitte beachten Sie, daß auf dem CE-516P das Zeichen "a" in der Programmlisting als "a" und nicht als "a" gedruckt wird.

## PROGRAMMTITEL: Umwandlung zwischen orthogonalen Kordinaten und Polarkoordinaten

Dies ist ein sehr nützliches Programm für die Umwandlung zwischen dreidimensionalen orthogonalen Koordinaten und Polarkoordinaten in beiden Richtungen. Die Winkeleinheit bei Dateneingabe und Ausgabe der Ergebnisse entspricht der Voreinstellung.

### ■ BEDIENUNG

1. **DEF A** (Umwandlung zu Polarkoordinaten von orthogonalen Koordinaten) Wenn nach Abruf auf dem Display die Koordinaten  $x, y, z$  eingegeben werden, erscheinen die Werte für  $r, \theta, \varphi$  auf dem Display in dieser Reihenfolge, danach endet das Programm.
2. **DEF B** (Umwandlung zu orthogonalen Koordinaten von Polarkoordinaten) Wenn nach Abruf auf dem Display die Koordinaten  $r, \theta, \varphi$  eingegeben werden, erscheinen die Werte für  $x, y, z$  auf dem Display in dieser Reihenfolge, danach endet das Programm.

Hinweis) Bei Voreinstellung der Winkeleinheit als DEG sind Eingabe und Ausgabe in Altgrad und bei Voreinstellung RAD in Radian.

### ■ PROGRAMMINHALT

#### 1. Umwandlung zu Polarkoordinaten von orthogonalen Koordinaten

mit  $r = y = 0, r = 0, \theta = \text{unbestimmt}$

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = \sin^{-1}(z/r)$$

mit  $x > 0, \varphi = \tan^{-1}(y/x)$

mit  $x = 0, \text{ wenn } y \geq 0, \varphi = 90^\circ$

    wenn  $y < 0, \varphi = 270^\circ$

mit  $x < 0, \text{ wenn } y \geq 0, \varphi = \tan^{-1}(y/x) + 180^\circ$

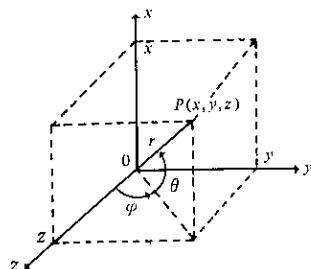
    wenn  $y < 0, \varphi = \tan^{-1}(y/x) - 180^\circ$

#### 2. Umwandlung zu orthogonalen Koordinaten von Polarkoordinaten.

$$x = r \cos \theta \cdot \cos \varphi$$

$$y = r \cos \theta \cdot \sin \varphi$$

$$z = r \sin \theta$$



## ■ BEISPIEL

### 1. Umwandlung von orthogonalen Koordinaten zu Polarkoordinaten

$$x = -1$$

$$y = 2$$

$$z = -3$$

Winkeleinheit DEG

### 2. Umwandlung von Polarkoordinaten zu orthogonalen Koordinaten

$$r = 3.741657387$$

$$\theta = -53.30077479$$

$$\varphi = 116.5650512$$

Winkeleinheit DEG

## ■ TASTENBETÄIGUNG

< Orthogonale Koordinaten → Polarkoordinaten >

1. **DEF A**

2. -1 **ENTER**

3. 2 **ENTER**

4. -3 **ENTER**

5. **ENTER**

6. **ENTER**

7. **ENTER**

8. **ENTER**

9. **ENTER**

10. **ENTER**

< Polarkoordinaten → Orthogonale Koordinaten >

1. **DEF B**

2. 3.741657387 **ENTER**

3. -53.30077479 **ENTER**

4. 116.5650512 **ENTER**

5. **ENTER**

6. **ENTER**

7. **ENTER**

8. **ENTER**

9. **ENTER**

10. **ENTER**

## ■ PROGRAMMLISTING

```

10:"A" CLEAR
20:INPUT "x=";X
30:INPUT "y=";Y
40:INPUT "z=";Z
50:R=SQR (X*X+Y*Y+Z*Z);
60:IF R=0 WAIT :PRINT "r=0 Undefined":END
70:C=ASN (Z/R);IF X>0,C:=C
80:LET F=ATN (Y/X);GOTO 90
90:PRINT R;PRINT "THETA";
   PRINT C;PRINT "PHI"
   PRINT F:END
100:"B" CLEAR
110:INPUT "r=";R
120:INPUT "THETA=";C
130:INPUT "PHI=";F
140:Y=R*COS C:X=Y*COS F;
    Y=Y*SIN F:Z=R*SIN C
150:WAIT :PRINT "x";
    PRINT X:PRINT "y";
    PRINT Y:PRINT "z";
    PRINT Z:END

```

311 bytes

## ■ SPEICHERINHALT

A	$\sqrt{x^2+y^2+z^2}$
C	$\theta$
F	$\varphi$
R	$r$
X	x-Koordinate
Y	y-Koordinate
Z	z-Koordinate

Dieses Programm dient zur Berechnung und Darstellung exponentieller Funktionen, die beispielsweise als Beziehung zwischen Zeit und Temperaturabfall oder zwischen Zeit und Kernspaltung in der Natur vorkommen. Nach Eingabe der Daten  $x, y$  berechnet dieses Programm die Koeffizienten  $a, b$  und den Korrelationskoeffizienten  $r$  nach der Formel  $y = ab^x$  und plottet die Kurve.

## ■ BEDIENUNG

1. **DEF** **A** (Starten des Programms.)
2. Nach Abfrage auf dem Display die Anzahl der Daten (N) eingeben. Danach entsprechend der Anzahl der Daten die Daten  $x, y$  eingeben, dann werden die Koeffizienten  $a, b$  und der Korrelationskoeffizient  $r$  berechnet und auf den Drucker ausgegeben.
3. Nach Betätigung von **DEF** **B** wird die Exponentalkurve in grün auf dem Drucker ausgegeben. Die eingegebenen Daten werden in blauer Farbe geplottet.
4. Wenn als nächstes der Schätzwert eingegeben wird, werden die eingegebenen Daten in roter Farbe geplottet und der Schätzwert  $x, y$  ausgedruckt. Danach endet das Programm.

Hinweis: Für die Anzahl der Daten bestehen die folgenden Grenzen.

Standardkapazität:	57
CE-201M	185
CE-202M	255

## ■ PROGRAMMINHALT

Die Punkte  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  entsprechend der Anzahl der Daten  $n$  werden mit der Exponentialfunktion  $y = a \cdot b^x$  berechnet. Die Positionen  $Y = l_n y, A = l_n a, B = l_n b$  werden erhalten und dann zum linearen Ausdruck  $Y = A + Bx$  umgewandelt. A und B werden nach der Methode der kleinsten Quadrate wie folgt berechnet.

$$B = \frac{\sum x_i \cdot Y_i - n \cdot \bar{x} \cdot \bar{Y}}{\sum x_i^2 - n \cdot \bar{x}^2}, \quad A = \bar{Y} - B \cdot \bar{x}$$

$$\text{mit: } Y_i = l_n y_i, \quad \bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i, \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Sobald A und B bestimmt sind, werden  $a$  und  $b$  durch  $a = e^A$  und  $b = e^B$  errechnet. Weiterhin wird der Korrelationskoeffizient  $r$  von  $x$  und  $y$  berechnet.

## ■ BEISPIEL

x	0.5	1.2	3.1	7.4
y	7.01	11.72	44.54	936.71

- \* Wenden Sie die obigen Daten auf  $y = a \cdot b^x$  an und schätzen den Wert mit  $x = 2, 4, 6, 6.5$ .

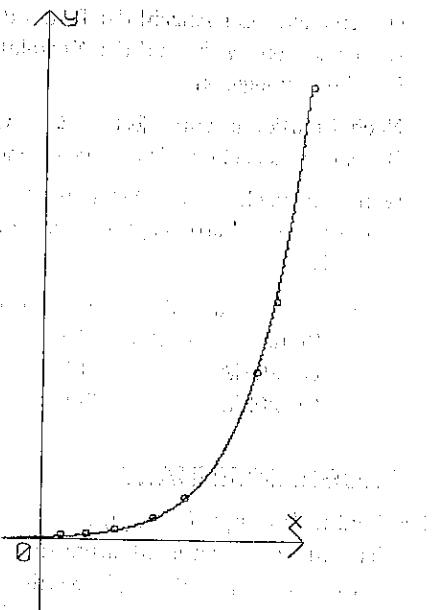
## ■ AUSDRUCK

(Ausdruck in Farbe. Siehe Seite I.)

$$n=9.999842365E-01$$

$$a=4.960331916$$

$$b=2.03057723$$



### \* Estimation \*

$$\begin{array}{ll} x=2 & y=20.45265825 \\ x=4 & y=84.3312981 \\ x=6 & y=342.7185094 \\ x=6.5 & y=495.4930476 \end{array}$$

## ■ TASTENBETÄIGUNG

< Dateneingabe >

1. **[DEF]** **[A]**

2. 4 **[ENTER]**

3. 0.5 **[ENTER]**

4. 7.01 **[ENTER]**

5. 1.2 **[ENTER]**

6. 11.72 **[ENTER]**

⋮

Gleiche Eingabe wie oben.

⋮

7. 936.71 **[ENTER]**

< Ausgabe der Ergebnisse >

1. **[DEF]** **[B]**

2. 2 **[ENTER]**

3. 4 **[ENTER]**

4. 6 **[ENTER]**

5. 6.5 **[ENTER]**

## ■ PROGRAMMLISTING

```

10;"a" CLEAR : WAIT 0; INPUT "N=";N
20:DIM X(N-1),Y(N-1),XS(N-1),YS(N-1):E
=100#D=-E:F=0
30:FOR I=0 TO N-1
40:PAUSE "x"+STR$ (I+1)+"=": INPUT X
(I): GOTO 80
50:N=I: GOTO 110
60:PAUSE "y"+STR$ (I+1)+"=": INPUT Y
(I)
70:Y=LN Y(I): IF DXX(I) LET D=DX(I)
80:IF E>X(I) LET E=X(I)
90:IF F<Y(I) LET F=Y(I)
100:D=0:X(I):P=P+Y(I)*D*X(I):R=R+Y*
Y:S=S+X(I)*Y: NEXT I
110:X=0:N=P/N:T=Q-NXXX:U=S-NXYX:U=R
-NXYX
120:C=U/SQR (T*U):B=U/T:A=EXP (-Y*B*X):B
=EXP B
130:GOSUB 380
140:LPRINT "x";C: LPRINT "a";A:
LPRINT "b";B: CLOSE : END
150:"B"=M=300: GOSUB 380
160:IF E>0 LET Z=25:L=D/175: GOTO 180
170:L=(D+ABS E)/200:Z=ABS E/L+5
180:LPRINT CHR$ 27;"b": LPRINT "M"; INT
Z+2301,-223": LPRINT "1": LPRINT "
L0"
190:LPRINT "M";- INT Z;"0": LPRINT "D"
;200- INT Z;"0";"200- INT Z-10";,-
10"
200:LPRINT "D";200- INT Z;"0";0,-200-
INT Z-10;"10": LPRINT "Px"
210:LPRINT "M0,-50": LPRINT "00,350,-10
,340,0,350,10,340": LPRINT "Py"
220:LPRINT "M-15,-15": LPRINT "Po"
230:LPRINT CHR$ 27;"1": FOR I=0 TO N-1:
J=X(I)/L:K=Y(I)/M: GOSUB 370: NEXT
I
240:LPRINT CHR$ 27;"2";J=-Z: LPRINT "M"
; INT J;""; INT (ARB*XJ*L)/M
250:J=J+2: IF J>200-2 GOTO 270
260:K=A*X(B*(J*L))/M: IF K<350 LPRINT "D"
; INT J;""; INT K: GOTO 250
270:I=0: LPRINT CHR$ 27;"3"
280:INPUT "Est. x=";XS(I): GOTO 300
290:GOTO 320
300:J=XS(I)/L:Y(S(I))=A*X(S(I))/M
: IF K<350 GOSUB 370
310:I=I+1: IF I>N THEN 280
320:LPRINT "M0,-50": LPRINT CHR$ 27;"a"
: LPRINT CHR$ 27;"0": IF I=0 THEN 3
62
330:LPRINT CHR$ 27;"b": LPRINT "* Esti
mation"
340:FOR H=0 TO I-1: LPRINT USING "~~~~~
~~~~~";"x"+STR$ XS(H):
350:LPRINT "y"+STR$ YS(H): NEXT H
360:LPRINT USING "": LPRINT "": LPRINT
"": CLOSE : END
370:LPRINT "M"; INT J;""; INT K-1:
LPRINT CHR$ 27;"Po": LPRINT "Po":
RETURN
380:CLOSE : OPEN "1200,N,8,1,A,L":
CONSOLE 100: LPRINT CHR$ 27;"a"
390:LPRINT CHR$ 27;"b": LPRINT CHR$ 27
;"0": RETURN

```

## ■ SPEICHERINHALT

A	<i>a</i>
B	<i>b'</i> , <i>b</i>
C	Korrelationskoeffizient <i>r</i>
D	X - MAX
E	X - MIN
F	Y - MAX
H	✓
I	✓
J	✓
K	✓
L	✓
M	✓
N	Anzahl der Daten
O	$\Sigma xi$
P	$\Sigma Yi$
Q	$\Sigma x^2 i$
R	$\Sigma Y^2 i$
S	$\Sigma xi \cdot Yi$
T	$S_{xx}$
U	$S_{xy}$
V	$S_{yy}$
X	$\bar{x}$
Y	$\bar{y}$
Z	✓
X(N-1)	X-Daten
XS(N-1)	Schätzwert für X
Y(N-1)	Y-Daten
YS(N-1)	Schätzwert für Y

1200 bytes

Für Qualitätskontrolle bei der Herstellung von Waren müssen zuerst die Faktoren gesucht werden, die die Qualität beeinflussen. Dann werden natürlich zuerst die Faktoren geändert, die den stärksten Einfluß haben. Diese Faktoren, die einen größeren Einfluß als andere haben, können mit dem Pareto-Diagramm gesucht werden. Dieses Programm, listet verschiedene Faktoren in der Reihenfolge ihrer Häufigkeit und druckt die Daten in einer einfach verständlichen Grafik aus.

## ■ BEDIENUNG

1. **DEF A** (Programmstart)
2. Entsprechend der Abfrage auf dem Display für alle Faktoren nacheinander die Namen der Faktoren (bis zu 8 Zeichen) und die Anzahl der Defekte eingeben. Wird nur **ENTER** gedrückt, ist die Eingabe beendet. Dann geht das Programm weiter zu Schritt 3.
3. In diesem Schritt können die eingegebenen Daten korrigiert werden.
  - Wenn Korrektur erforderlich ist - - - **Y ENTER**
  - Keine Korrektur erforderlich - - - **N ENTER**(Wenn keine Korrektur gewünscht wird, geht das Programm weiter zu Schritt 4.)  
\* Korrektur der Daten  
Die Namen der Faktoren und die Anzahl der Defekte werden nacheinander angezeigt. Wenn eine Korrektur erforderlich ist, den Namen bzw. die Anzahl der Defekte erneut eingeben, sonst **ENTER** drücken. Nach einem Durchgang aller Faktoren muß erneut entschieden werden, ob noch eine weitere Korrektur erforderlich ist.
4. Nach Abfrage auf dem Display den Titel (bis zu 10 Zeichen) und das Datum eingeben. Dann erfolgt die Berechnung und wird das Pareto-Diagramm auf dem Drucker ausgegeben. Danach endet das Programm.

Hinweis) Die Anzahl der Faktoren kann durch Ändern des Wertes A = 18 in Zeile 10 modifiziert werden.

CE-201M 149 Faktoren  
CE-202M 255 Faktoren

## ■ BEISPIEL

Das folgende Beispiel ist eine Bewertung der Anzahl defekter Teile der einzelnen Maschinen einer Elektronik-Fabrik. Das Datum ist der 15.2.1985, als Titel wurde "Pareto" gewählt.

No.	Faktorname	Daten
1	Maschine A	10
2	Maschine B	20
3	Maschine C	15
4	Maschine D	30
5	Maschine E	40
6	Maschine F	80
7	Maschine G	100
8	Maschine H	90

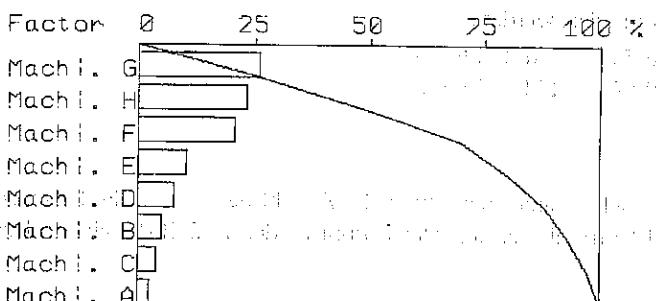
## AUSDRUCK

(Ausdruck in Farbe. Siehe Seite I.)

### Pareto

DWG date: 1985/2/15

No.	Factor	Defects	Ratio %
1	Machi. G	100	25.92
2	Machi. H	90	23.37
3	Machi. F	80	20.72
4	Machi. E	40	10.38
5	Machi. D	30	7.79
6	Machi. B	20	5.19
7	Machi. C	15	3.89
8	Machi. A	10	2.59



## ■ TASTENBETÄIGUNG

1. **DEF** **A**

\* Pareto Diagram \*

Fact. name (1)

= \_

2. M. **SPC** A **ENTER**

No. of def. (1)

= \_

3. 10 **ENTER**

Fact. name (2)

= \_

:

Gleiche Eingabe wie oben.

:

4. 90 **ENTER**

Fact. name (9)

= \_

7. **ENTER**

10

->

8. **ENTER**

(2) M. B

->

9. **ENTER**

15

->

10. 20 **ENTER**

(3) M. B

->

:

Die Daten überprüfen und  
gegebenenfalls korrigieren.

:

11. **ENTER**

Correct? (Y/N) \_

12. **N** **ENTER**

Title = \_

13. P **SML** ARETO **ENTER**

Year = 19\_

14. 85 **ENTER**

Month =

15. 2 **ENTER**

Day =

16. 15 **ENTER**

>

## ■ PROGRAMMLISTING

```
10;"A" USING : CLEAR : WAIT 0:A=B=8
20:PAUSE "*Pareto Diagramm"
30:DIM B$(A)B$,C(A),D(A),Z$(0)*B,N$(0)
  *10:U$=CHR$ 27
40:FOR I=1 TO A: PAUSE "Fact. name (<"; STR$ I;">"
50:BEER 1: INPUT "=";Z$(0): GOTO 90
60:C=I-1: IF C<1 BEEP 2: GOTO 10
70:IF C<2 BEEP 2:I=2: GOTO 50
80:I=A: GOTO 160
90:IF LEN Z$(0)>8 THEN 50
100:B$(I)=Z$(0)
110:PAUSE "No. of def. (<";STR$ I;">"
120:BEER 1: INPUT "=";Z$(0)
130:C(I)=VAL Z$(0)
140:IF C(I)<#0 BEER 2: GOTO 120
150:E=E+C(I):C=I:Z$(0)=""
160:NEXT I
170:BEEP 1: INPUT "Correct? (Y/N)":Z$:Z
  :=CHR$ (ASC Z$ AND 233)
180:IF Z$="N" GOSUB 490: GOTO 320
190:IF Z$<>"Y" THEN 120
200:FOR I=1 TO C
210:WAIT 150: PRINT "(";STR$ I;")":B$(I)
  )
220:INPUT " -> ";Z$(0): GOTO 240
230:GOTO 260
240:IF LEN Z$(0)>8 BEEP 2: GOTO 210
250:B$(I)=Z$(0)
260:PRINT STR$ C(I)
270:INPUT " -> ";Z$(0): GOTO 290
280:GOTO 310
290:IF VAL Z$(0)<=0 BEEP 1: GOTO 260
300:E=E-C(I):C(I)=VAL Z$(0):E=E+C(I)
310:NEXT I: GOTO 120
320:CONSOLE 90: OPEN "1200:N,B1,A,L";
  GOSUB 810:H=0:U=0: GOSUB 700
330:LPRINT CHR$ 13: LPRINT U$;"a";U$;"?
  b"
340:H=0:U=0: LPRINT : LPRINT
350:LPRINT "Factor 0    25      50
  75    100 %"
360:LPRINT U$;"b";"R95,15": LPRINT "I"
370:LPRINT "D340:0,340,";C*-24;"0;"C*
  "-24;"0,0"
380:FOR I=1 TO 3: LPRINT "R";25*3.45^I,0
390:LPRINT "J0,5,0,-5": NEXT I
```

```
400:LPRINT "H": LPRINT "R0,-6": LPRINT
  U$;"3"
410:FOR I=1 TO C
420:LPRINT "J";(D(I)-U)*3.4;"0,0,-18,";
  ;U-(D(I))*3.4;"0"
430:LPRINT "R0,-6":U=D(I): NEXT I
440:LPRINT "H": LPRINT U$;"2"
450:FOR I=1 TO C: LPRINT "O";D(I)*3.4;"0"
  ;"I*-24: NEXT I
460:LPRINT "H": LPRINT U$;"2"
470:FOR I=1 TO C: LPRINT "M-55,";I*-24:
  LPRINT "P": USING "8888&888":B$(I):
  USING : NEXT I
480:LPRINT U$;"a": LPRINT : LPRINT :
  USING : END
490:L= INT (C/2+1):R=C
500:IF L>1 LET L=L-1:X=C(L):B$(0)=B$(L)
  : GOTO 530
510:X=C(R):B$(0)=B$(R):C(R)=C(1):B$(R)=
  B$(1):R=R-1
520:IF R>=1 LET C(1)=X:B$(1)=B$(0):
  GOTO 600
530:J=L
540:I=J:J=2*I
550:IF J>R THEN 590
560:IF J<R THEN IF C(J)>C(J+1) LET J=J+
  1
570:IF X<=C(J) THEN 530
580:(I)=C(J):B$(I)=B$(J): GOTO 540
590:(I)=X:B$(I)=B$(0): GOTO 500
600:FOR I=1 TO C:H=H+C(I):D(I)=H/E*100:
  NEXT I: RETURN
610:INPUT "Title=";IN$(0)
620:IF LEN IN$(0)>10 BEEP 2: GOTO 610
630:CC$="999": INPUT "Year =13";CC$
640:IN$=1900+ INT VAL CC$: IF N>1999 BEEP
  1: GOTO 630
650:DD$="99": INPUT "Month =";DD$:
660:D= INT VAL DD$: IF D>12 BEEP 2:
  GOTO 650
670:EE$="99": INPUT "Day ="EE$:
680:F1= INT VAL EE$: IF F1>31 THEN 670
690:RETURN
700:LPRINT : LPRINT U$;"?c";U$;"b";U$;"0"
  "

```

```

210:LPRINT "M100,0": LPRINT "P$LEFT$(  

    NF(0),10)  

220:LPRINT US$"P": LPRINT "M100,-60"  

230:LPRINT "PONG date: 1514-01-01"  

    STR$ DI$"/1514-01-01"  

240:LPRINT "R80,-60": LPRINT "M10,-130"  

    : LPRINT "I"  

250:I=1: FOR I=0 TO I+1  

260:LPRINT "J"IF#(443),0"  

270:LPRINT "R0,-30": F=F+1: NEXT I:  

    LPRINT "M0,"I-1,I+1,30  

280:RESTORE :I-1: FOR I=1 TO I  

290:LPRINT "J0,"I*30*(I+1): READ W  

300:LPRINT "P"100*121"2"IF F> 1: NEXT I  

310:LPRINT "M12,-20": LPRINT "F"1"R0,"  

    LPINT "R30,0"  

320:LPRINT "PFacile"  

330:LPRINT "R80,0": LPRINT "PDefect":  

    LPINT "R80,0"  

340:LPRINT "PRefo %"  

350:FOR I=1 TO C  

360:LPRINT "F10,"$1*-30-W: LPRINT "P"  

    USING "###";I  

370:LPRINT US$"0": LPRINT "P24,0"  

380:USING : LPRINT "P$LEFT$(B$(I),6):  

    LPRINT "M100,"I-130-20  

390:LPRINT "P": LSING "######";CC$)  

400:LPRINT "R48,0": LPRINT USING "####.  

    #####";P$;D(I)-U: USING :U*D(I)  

410:NEXT I: RETURN  

420:DATA 4,11,13,9,8

```

2204 bytes

## ■ SPEICHERINHALT

A	Anzahl der Faktoren
B	✓
C	✓
D	Monat
E	gesamt
F	Flag
H	✓
I	✓
J	✓
L	✓
N	Jahr
R	Endzeiger
V	✓
W	für Datenlesen
X	✓
C(A)	Anzahl der Defekte
D(A)	Zusammensetzungeverhältnis
F1	Tag
U\$	ESC(CHR\$27)
Z\$	✓
N\$(0)	✓
Z\$(0)	für Eingabe
B\$(A)	Faktorname
CC\$	für Eingabe des Jahres
DD\$	für Eingabe des Monats
EE\$	für Eingabe des Tages

Polygone lassen sich immer in mehrere Dreiecke zerlegen. Daher bietet sich an, zur Berechnung der Fläche eines Polygons die Flächen der einzelnen Dreiecke zu berechnen und zu addieren.

## ■ BEDIENUNG

### 1. **DEF A** (Programmstart)

Entsprechend der Abfrage auf dem Display die Anzahl der Eckpunkte und die Koordinaten ( $x, y$ ) jedes Eckpunktes eingeben.

### 2. **DEF B** betätigen.

Auf dem Display werden die Nummern von drei Eckpunkten abgefragt. Wird nach dieser Eingabe **ENTER** gedrückt, wird die Fläche des Dreiecks, daß durch die drei eingegebenen Eckpunkte gebildet wird, auf dem Drucker ausgegeben.

### 3. Durch Betätigung von **DEF C** wird die Gesamtfläche gelöscht, dann endet das Programm.

Hinweis) Für die Anzahl der Eckpunkte gelten die folgenden Grenzen.

Standardkapazität: 129 Eckpunkte

CE-201M, CE-202M: 255 Eckpunkte

## ■ PROGRAMMINHALT

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

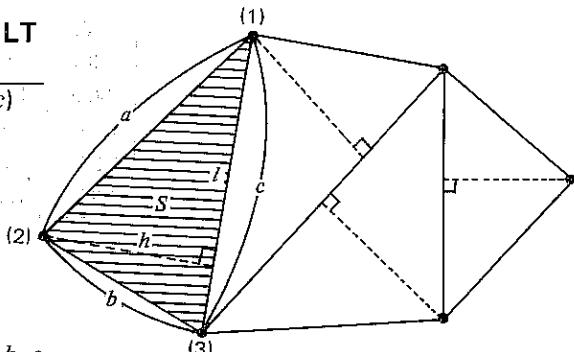
$$s = \frac{a+b+c}{2}$$

Fläche des Dreiecks

$$S' = \frac{1}{2} h L$$

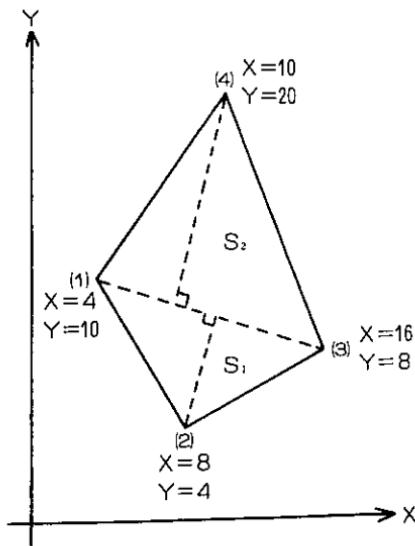
L: Längste Seite unter  $a, b, c$

H: Die Werte können bis zu vier Nachkommastellen eingegeben werden.



## ■ BEISPIEL

Berechnen Sie die Fläche des folgenden Polygons.



## ■ AUSDRUCK

Point 1x= 4  
Point 1y= 10  
Point 2x= 8  
Point 2y= 4  
Point 3x= 16  
Point 3y= 8  
Point 4x= 10  
Point 4y= 20

1-2-3

A= 7.211  
B= 8.944  
C= 12.166  
H= 5.260  
S= 31.996580

1-3-4

A= 12.166  
B= 13.416  
C= 11.662  
H= 9.838  
S= 65.993304

\*TS\*= 97.989884

## ■ TASTENBETÄIGUNG

< Eingabe der Koordinaten >

1. [DEF] [A]

Numbers =

5. 1 [ENTER]

Point =

2. 4 [ENTER]

Point 1x =

6. 3 [ENTER]

Point =

?

7. 4 [ENTER]

Point =

3. 4 [ENTER]

Point 1y =

8. [ENTER]

?

>

Gleiche Eingabe wie oben.

< Löscht die Gesamtfläche >

1. [DEF] [C]

\* \* TS CLEAR \* \*

>

< Eingabe der Koordinaten-Nummern >

1. [DEF] [B]

Point =

2. 1 [ENTER]

Point =

3. 2 [ENTER]

Point =

4. 3 [ENTER]

Point =

## ■ PROGRAMMLISTING

```

10:"A": USING :CLEAR :
    WAIT 0
20:INPUT "Numbers=";N
30:IF N<1 BEEP 2:GOTO 2
    0
40:DIM X(N-1),Y(N-1),B$(0)
50:FOR I=0 TO N-1
60:B$(0)="x":GOSUB 360
70:INPUT X(I);B$(0)="y=";
    "+STR$ X(I):GOSUB 3
    70:GOTO 90
80:N=I:END
90:B$(0)="y":GOSUB 360
    :INPUT Y(I)
100:B$(0)="y=";STR$ Y(I)
    ;:GOSUB 370:NEXT I
110:BEEP 1:END
120;"P" USING :INPUT "Point=";P;
    Point=";P:GOTO 140
130:GOTO 310
140:IF (0<1)+(0>N)+(P<1)
    +(P>N)+(Q<1)+(Q>N)<>
    0 GOTO 120
150:C=X(0-1):D=Y(0-1):E=
    X(P-1):F=Y(P-1):G=X(Q-1):
    H=Y(Q-1)
160:X=E-C:Y=F-D:GOSUB 33
    0
170:A=X:K=G-E:Y=H-F:
    GOSUB 330
180:B=X:C=G:Y=D-H:
    GOSUB 330
190:C=X:IF A>X LET X=A
200:IF B>X LET X=B
210:I=(A+B+C)/2:S=SQR (I
    *(I-A)*(I-B)*(I-C))
220:J= INT ((2*5/X)*10^3
    )/10^3:L=X:GOSUB 340
230:X=L:S=X*J/2:K=K+S:L=
    A:GOSUB 340
240:A=L:L=B:GOSUB 340
250:B=L:L=C:GOSUB 340
260:C=L:L=S:GOSUB 350
270:S=L:L=K:GOSUB 350
280:K=L:L=LPRT "
    0+""+STR$ P+"-"
    STR$ Q
290:LPRT "A="; USING
    "#####";A

```

959 bytes

## ■ SPEICHERINHALT

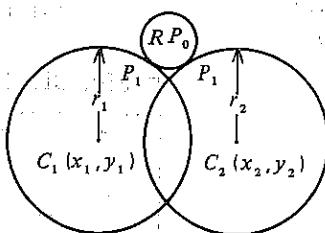
A	<i>a</i>
B	<i>b</i>
C	<i>x</i> <sub>1</sub> , ✓
D	<i>y</i> <sub>1</sub>
E	<i>x</i> <sub>2</sub>
F	<i>y</i> <sub>2</sub>
G	<i>x</i> <sub>3</sub>
H	<i>y</i> <sub>3</sub>
I	S
J	<i>h</i>
K	$\Sigma s$
L	✓
N	Anzahl der Eckpunkte
O	✓
P	✓
Q	✓
S	S
X	✓
Y	✓
X(N-1)	<i>x</i> -Koordinaten
Y(N-1)	<i>y</i> -Koordinaten
B\$(0)	✓

# PROGRAMMTITEL: Anschmiegung eines Kreises an zwei Kreise

Dieses Programm lockert die strenge Geometrie etwas auf. Es berechnet die Mittelpunktkoordinaten eines Kreises, das sich an zwei Kreise anschmiegt, und die Koordinaten der Tangentenpunkte der zwei Kreise. Eingegeben werden müssen die Mittelpunktkoordinaten der zwei Kreise, die Radien aller drei Kreise und drei Unterscheidungskriterien.

## ■ BEDIENUNG

1. **DEF A** (Programmstart)
2. Entsprechend der Abfrage auf dem Display die Mittelpunktkoordinaten  $(x_1, y_1)$  und den Radius  $r_1$  von Kreis  $C_1$ , die Mittelpunktkoordinaten  $(x_2, y_2)$  und den Radius  $r_2$  von Kreis  $C_2$  und den Radius  $r$  des Kreises, der sich an  $C_1$  und  $C_2$  anschmiegt, eingeben.

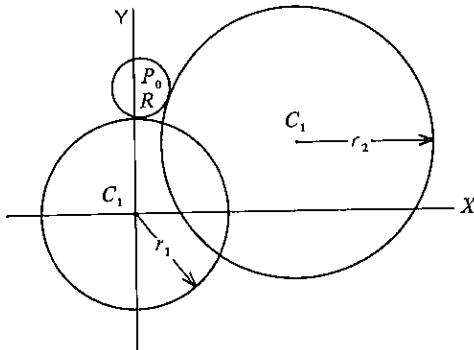


3. Dann die Daten für die Unterscheidungskriterien eingeben.  
(Unterscheidungskriterien)

(1)	Wenn der Kreis extern Tangente zu Kreis $C_1$ ist.	1
	Wenn der Kreis intern Tangente zu Kreis $C_1$ ist.	-1
(2)	Wenn der Kreis extern Tangente zu Kreis $C_2$ ist.	1
	Wenn der Kreis intern Tangente zu Kreis $C_2$ ist.	-1
(3)	Wenn sich der Kreis an der linken Seite befindet, gesehen von der Mitte von Kreis $C_1$ auf Kreis $C_2$ .	1
	Wenn sich der Kreis an der rechten Seite befindet, gesehen von der Mitte von Kreis $C_1$ auf Kreis $C_2$ .	-1

Nach Eingabe der Daten werden die Mittelpunktkoordinaten des Kreises, der sich an  $C_1$  und  $C_2$  anschmiegt, und die Koordinaten der Tangentenpunkte  $P_1$  und  $P_2$  auf dem Display angezeigt. Danach endet das Programm.

## ■ BEISPIEL



$$C_1: x_1 = 0, y_1 = 0, r_1 = 30$$

$$C_2: x_2 = 50, y_2 = 20, r_2 = 40$$

$$R = 10$$

Unterscheidungskriterien:

(1) 1 (Tangente extern)

(2) 1 (Tangente extern)

(3) 1 (linke Seite)

## ■ TASTENBETÄIGUNG

1. **DEF A**

**C1 x = \_**

7. 1 **ENTER**

**PØ x = 4.08**

2. 0 **ENTER**

**C1 y = \_**

8. **ENTER**

**PØ y = 39.79**

3. 0 **ENTER**

**C1 r = \_**

9. **ENTER**

**P1 x = 3.06**

:

Gleiche Eingabe wie oben.

:

4. 10 **ENTER**

**C1 . OUT : 1 IN : -1 = \_**

10. **ENTER**

**P1 y = 29.84**

11. **ENTER**

**P2 x = 13.27**

5. 1 **ENTER**

**C2 . OUT : 1 IN : -1 = \_**

12. **ENTER**

**P2 y = 35.83**

6. 1 **ENTER**

**LEFT : 1 RIGHT : -1 = \_**

13. **ENTER**

**> \_**

## ■ PROGRAMMLISTING

```

10:"A" DEGREE :INPUT "C
    1 x=";A
20:INPUT "C1 y=";B
30:INPUT "C1 r=";I
40:INPUT "C2 x=";D
50:INPUT "C2 y=";E
60:INPUT "C2 r=";P
70:INPUT "R=";S
80:INPUT "C1.DUT:1 IN:-1=";O
90:INPUT "C2.DUT:1 IN:-1=";R
100:INPUT "LEFT:1RIGHT:-1=";G
110:F=P+R*S:C=0+Q*S:H=0-
    A:I=E-B:J=SQR ((H*H+I*I)):K=ACOS (H/J):IF O>Y LET W=-W
    >I LET K=-K
120:L=ACOS ((C+C+J*J-F*F)/2/C/J)
130:N=K+G*L:M=A+C*COS N:
    N=B+C*SIN N:X=Q*(A-M):
    Y=Q*(B-N):GOSUB 24
    @
140:IF ((Q=-1)*(S>0))=1
    LET W=W+180
150:H=M+S*COS W:I=N+S*SIN W:X=R*(D-M):Y=R*(E-N):GOSUB 240
160:IF ((R=-1)*(S>P))=1
    LET W=W+180
170:J=M+S*COS W:K=N+S*SIN W
180:M=M+SGN M*.005:N=N+SGN N*.005
190:H=H+SGN H*.005:I=I+SGN I*.005
200:J=J+SGN J*.005:K=K+SGN K*.005
210:PRINT "P0 x="; USING "######
    "#";M:
    PRINT "P0 y=";N
220:PRINT "P1 x=";H:
    PRINT "P1 y=";I
230:PRINT "P2 x=";J:
    PRINT "P2 y=";K:END
240:W=ACOS (X/SQR (X*X+Y*Y)):IF O>Y LET W=-W
250:RETURN

```

651 bytes

## ■ SPEICHERINHALT

A	$x_1$
B	$y_1$
C	✓
D	$x_2$
E	$y_2$
F	✓
G	Unterscheidungskriterium 3
H	$P_1x$
I	$P_1y$
J	$P_2x$
K	$P_2y$
L	✓
M	$P_0x$
N	$P_0y$
O	$r_1$
P	$r_2$
Q	Unterscheidungskriterium 1
R	Unterscheidungskriterium 2
S	R
W	✓
X	✓
Y	✓

“Frequenzgang-Kurve” ist ein gut bekannter Begriff unter Audio-Enthusiasten. Dieses Programm erzeugt den Frequenzgang mit der Leitwertfunktion und gibt dann eine einfach verständliche Kurve auf dem Drucker aus.

## ■ BEDIENUNG

### 1. **[DEF] [A]** (Programmstart)

Der Rahmen der Kurve wird auf dem Drucker ausgegeben.

### 2. Entsprechend der Abfrage auf dem Display die Daten für Verstärkungsstufe (Steigungswert), maximale Verstärkung und minimale Verstärkung eingeben. Der Maßstab der Kurve wird auf dem Drucker ausgegeben.

### 3. Dann die Daten für Frequenzstufe, Grad der Leitwertfunktion, der Koeffizient des Nenners der Formel und der Koeffizient des Zählers der Formel eingeben. Die Kurve wird auf dem Drucker ausgegeben.

### 4. Als nächstes muß entschieden werden, ob die Daten der Koeffizienten zum Drucker ausgegeben werden oder ob eine weitere Leitwertfunktion berechnet werden soll.

- Ausdruck der Funktion ----- **[Y] [ENTER]**

(In diesem Fall endet das Programm nach Ausdruck der Funktion.)

- Berechnung der Leitwertfunktion ----- **[N] [ENTER]**

(In diesem Fall die gleichen Daten wie in Schritt 3 eingeben, dann wird die Kurve ausgedruckt. Danach muß wieder entschieden werden, ob die Daten der Funktion ausgegeben werden sollen oder die Leitwertfunktion berechnet werden soll.)

## ■ PROGRAMMINHALT

Formel

$$G(S) = \frac{B(0) + B(1)S + \dots + B(N-1)S^{N-1}}{A(0) + A(1)S + A(2)S^2 + \dots + A(N)S^N} \quad (N \leq 12)$$

dabei ist  $f$  (Hz) die Frequenz, aus der Formel folgt

$$\begin{aligned} G(j2\pi f) &= \frac{B(0) + B(1) \cdot j2\pi f + \dots + B(N-1) \cdot (j2\pi f)^{N-1}}{A(0) + A(1) \cdot j2\pi f + \dots + A(N) \cdot (j2\pi f)^N} \\ &= \frac{B(0) - B(2) \cdot (2\pi f)^2 + B(4) \cdot (2\pi f)^4 + \dots + j[B(1) \cdot 2\pi f - B(3) \cdot (2\pi f)^3 + \dots]}{A(0) - A(2) \cdot (2\pi f)^2 + \dots + j[A(1) \cdot 2\pi f - A(3) \cdot (2\pi f)^3 + \dots]} \end{aligned}$$

die Verstärkung  $|G|$  (dB) ergibt sich wie folgt

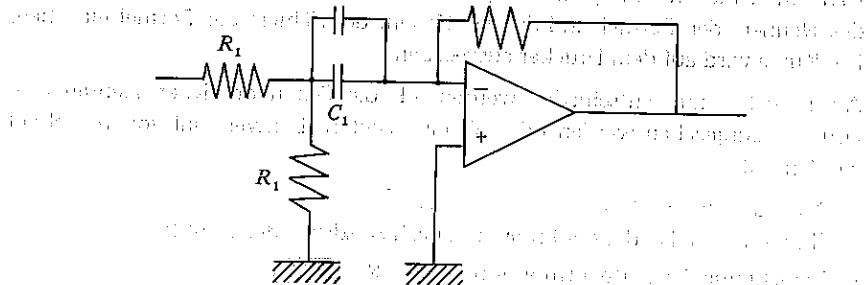
$\rightarrow G \approx 20 \log |G(j\omega)|$

$$\therefore G \approx 20 \log |G(j2\pi f)|$$

$$\begin{aligned} &= 20 \log \sqrt{\frac{|B(0) - B(2) \cdot (2\pi f)^2 + \dots|^2 + |B(1) \cdot 2\pi f - B(3) \cdot (2\pi f)^3 + \dots|^2}{|A(0) - A(2) \cdot (2\pi f)^2 + \dots|^2 + |A(1) \cdot 2\pi f - A(3) \cdot (2\pi f)^3 + \dots|^2}} \\ &= 10 \log \frac{|B(0) - B(2) \cdot (2\pi f)^2 + \dots|^2 + |B(1) \cdot 2\pi f - B(3) \cdot (2\pi f)^3 + \dots|^2}{|A(0) - A(2) \cdot (2\pi f)^2 + \dots|^2 + |A(1) \cdot 2\pi f - A(3) \cdot (2\pi f)^3 + \dots|^2} \end{aligned}$$

## ■ BEISPIEL

Der folgende Filter (Bandpassfilter) der Frequenzgang-Kurve wird auf dem Drucker gezeichnet.



$$G(S) = \frac{S/R_1 C_1}{S^2 + \frac{C_1 + C_2}{R_3 C_1 C_2} S + \frac{1}{R' R_3 C_1 C_2}}$$

$$R' \leq \frac{R_1 R_2}{R_1 + R_2}$$

Teilungswert der Verstärkung 25 (dB)

maximale Verstärkung 50 (dB)

minimaler Frequenzwert 1 (Hz)

mit:  $R_1 = 2k\Omega$ ,  $R_2 = 660\Omega$ ,  $R_3 = 100k\Omega$ ,

$C_1 = 0.01\mu F$ ,  $C_2 = 0.1\mu F$

Frequenzschritt = 80  $\Rightarrow$  Grad = 2

$$A(0) = 2015152, \quad B(0) = 0$$

$$A(1) = 1100, \quad B(1) = 50000$$

$$A(2) = 1$$

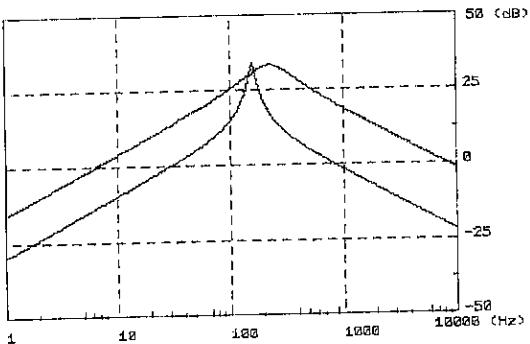
mit:  $R_1 = 2\text{k}\Omega$ ,  $R_2 = 660\Omega$ ,  $R_3 = 200\text{k}\Omega$   
 $C_1 = C_2 = 0.1\mu\text{F}$

Frequenzschritt = 80 : Grad = 2

$$\begin{array}{ll} A(0) = 1007576 & B(0) = 0 \\ A(1) = 100 & B(1) = 5000 \\ A(2) = 1 & \end{array}$$

## ■ AUSDRUCK

(Ausdruck in Farbe. Siehe Seite II.)



$$\begin{array}{ll} A(0)=1007576 & B(0)=0 \\ A(1)=100 & B(1)=5000 \\ A(2)=1 & \end{array}$$

## ■ TASTENBETÄIGUNG

1. **DEF A**  
GAIN STEP (dB) \_
2. **25 ENTER**  
GAIN MAX. (dB) \_
3. **50 ENTER**  
FREQ. MIN. (Hz) \_
4. **1 ENTER**  
FREQ. STEP = \_  
?
5. **80 ENTER**  
DEGREE 1 = \_  
?
6. **2 ENTER**  
A (0) = \_  
?
7. **2015152 ENTER**  
A (1) = \_  
?  
⋮  
Gleiche Eingabe wie oben.
8. **50000 ENTER**  
PRINTOUT Y/N \_
9. **N ENTER**  
FREQ. STEP = \_  
?  
⋮
10. **120 ENTER**  
DEGREE = \_  
?  
⋮
11. **2 ENTER**  
A (0) = \_  
?  
⋮
12. **1007576 ENTER**  
A (1) = \_  
?  
⋮
13. **100 ENTER**  
A (2) = \_  
?  
⋮
14. **5000 ENTER**  
PRINTOUT Y/N \_
15. **Y ENTER**  
> \_

## ■ PROGRAMMLISTING

```

18;"A" CONSOLE 160: OPEN "1200,N,B,1,A
,LT": USING :C=0: CLEAR
20:LPRINT ":" LPRINT CHR$ 2
;"\n"
30:LPRINT CHR$ 27;"b": LPRINT "10":
LPRINT CHR$ 27;"0"
40:LPRINT "M300,0": LPRINT "1": LPRINT
"0300,B,300,-200,B,-200,B,0"
50:LPRINT "L3"
60:LPRINT "M0,-50": LPRINT "D300,-50"
70:LPRINT "M300,-100": LPRINT "DB,-100
"
80:LPRINT "MB,-150": LPRINT "D300,-150
"
90:LPRINT "M225,-200": LPRINT "D225,0"
100:LPRINT "M150,0": LPRINT "D150,-200"
110:LPRINT "M25,-200": LPRINT "D25,0"
120:INPUT "GAIN STEP (dB)":B
130:INPUT "GAIN MAX. (dB)":V
140:INPUT "FREQ. MIN.(Hz)":SH
150:LPRINT "W": LPRINT CHR$ 27;"?a"
160:LPRINT "M300,-5": LPRINT "P"!STR$ P
;"(dB)"
170:FUR I=1 TO 4
180:LPRINT "M25,-";I-(25*(I-1)*5):
LPRINT "D300,":I-(25*(I-1)*5)
190:LPRINT "M205,0":I-1450: LPRINT "P":
STR$ (P-140): NEXT I
200:LPRINT "M205,-210": LPRINT "P"!STR$ (H*10^4*I)^(HZ)
210:FUR I=0 TO 3
220:LPRINT "M";I^K5;"-215": LPRINT "P"
;STR$ (H*10^4*I): NEXT I
230:FOR I=0 TO 3:S=40^I: FOR J=2 TO S:K
= INT (LOG (J*4^I))
240:LPRINT "M";K1;"-195": LPRINT "D";K1;
"-200": NEXT J: NEXT I
250:DIM A(15),B(15),W(15)
260:C=3:N=1
270:PAUSE "FREQ. STEP=: INPUT F
280:PAUSE "DEGREE=: INPUT H
290:FOR I=0 TO N: PAUSE "AC":STR$ I$=":
INPUT AC(I)
300:NEXT I
310:FUR I=0 TO N-1: PHASE "B":STR$ I$ =
;"": INPUT BC(I)
320:NEXT I
330:FOR I=N+1 TO 15:H(I)=B(I):C(I)=0: NE(X
)
340:B(I)=0
350:D=12
360:R=0:44: GOSUB 300: LPRINT "M0,"I-3:1
=302/F: LPRINT CHR$ 27;C(I);"\2"
370:FUR I=1 TO F: GOSUB 300: LPRINT "D"
; INT (I*4^I)^(HZ): G: NEXT I
380:LPRINT "M0,-210"
390:I=C-1: IF C>0 THEN 430
400:P="": INPUT "PRINT(LT, Y, H)":P
410:IF P<="Y" THEN GOTO 430
420:GOTO 210
430:LPRINT CHR$ 27;"W": LPRINT CHR$ 27
;"?a"
440:FUR I=0 TO N-1: LPRINT D(I):"DEEEX
XEEEEECEEEEEEK";" "
450:LPRINT "P"!STR$ R":P":= "+STR$ R01 :
```

1477 bytes

## SPEICHERINHALT

DRUCKER: DRAHAGG, 01.01.2011

A	reeller Teil des Nenners der Formel (1)
B	imaginärer Teil des Nenners der Formel (1)
C	Farbdefinition der Zeilen
D	reeller Teil des Zählers der Formel (1)
E	imaginärer Teil des Zählers der Formel (1)
F	Teilungswert der Zahlen der Kurve
G	Verstärkung (dB) zu $f$
H	minimaler Wert der Frequenz
I	Schleifenzähler
J	Schleifenzähler
K	Schleifenzähler
L	Schleifenzähler
M	Schleifenzähler
N	Grad der Leitwertfunktion
O	Teilungswert der Verstärkung
P	Maximalwert der Verstärkung
Q	Verstärkung zu $f$ (in Koordinaten umgewandelt)
R	Verstärkung zu $f$ (in Koordinaten umgewandelt)
S	$2\pi f$
T	✓
U	✓
V	✓
W	✓
X	✓
Z\$	für Tasteneingabe
A(15)	Koeffizient des Nenners der Formel (1)
B(15)	Koeffizient des Zählers der Formel (1)
W(15)	$(w_0)^t = (2\pi f)^t$

Das Kreisdiagramm ist sehr praktisch, um den Anteil eines Postens an der Gesamtmenge auf einem Blick zu ersehen. Dieses Programm kann ein Kreisdiagramm zeichnen, einen Abschnitt aus dem Kreisdiagramm herausnehmen und eine Tabelle der Daten ausgeben.

## ■ BEDIENUNG

1. **[DEF] [A]** (Programmstart)
2. Entsprechend der Abfrage auf dem Display den Titel des Diagramms, die Anzahl der Daten, die Posten und den Datentitel eingeben.
3. Danach die Daten eingeben. Die Posten-Nr. wird auf dem Display angezeigt. Den Namen des Postens und die Daten eingeben. Diesen Vorgang für alle Daten wiederholen.
4. In diesem Schritt ist Korrektur der eingegebenen Daten möglich.
  - Korrektur erforderlich: **[Y] [ENTER]**
  - Keine Korrektur erforderlich: **[N] [ENTER]**  
(Wenn keine Korrektur erforderlich ist, geht das Programm zu Schritt 5 weiter.)
- \* **Korrektur der Daten**  
Die Posten-Nr. der zu korrigierenden Daten und danach die Daten eingeben.  
Nach der Korrektur muß wieder entschieden werden, ob eine weitere Korrektur erforderlich ist.
5. In diesem Schritt können die eingegebenen Daten sortiert werden.
  - Sortieren gewünscht: **[Y] [ENTER]**
  - Sortieren nicht gewünscht: **[N] [ENTER]**
6. In diesem Schritt wird entschieden, ob das Kreisdiagramm gezeichnet wird.
  - Zeichnen des Kreisdiagramms: **[Y] [ENTER]**
  - Kreisdiagramm nicht zeichnen: **[N] [ENTER]**
7. In diesem Schritt wird entschieden, ob ein Abschnitt aus dem Kreisdiagramm herausgenommen werden soll.
  - Abschnitt herausnehmen: **[Y] [ENTER]**
  - Keinen Abschnitt herausnehmen: **[N] [ENTER]**  
(In diesem Fall macht das Programm mit Schritt 8 weiter.)
- \* **Bestimmung des Abschnitts, der herausgenommen werden soll.**  
Die Nr. der Daten eingeben, von denen der Abschnitt herausgenommen werden soll.

## 8. In diesem Schritt wird entschieden, ob eine Tabelle der Daten erstellt werden soll.

• Tabelle erstellen: **Y** **ENTER**

• Tabelle nicht erstellen: **N** **ENTER**

Nach Durchführung der obigen Schritte wird die Tabelle der Daten und das Kreisdiagramm oder nur das Kreisdiagramm auf dem Drucker ausgegeben, danach endet das Programm.

- Hinweis) • Titel des Kreisdiagramms – bis zu 10 Zeichen  
• Anzahl der Daten – von 2 bis zu 20  
• Posten, Titel der Daten – bis zu 10 Zeichen  
• Name für Posten – bis zu 10 Zeichen

Wird nach einem der oben dargestellten Befehle gefragt, so kann die Tabelle ausdruckt werden.

### ■ BEISPIEL

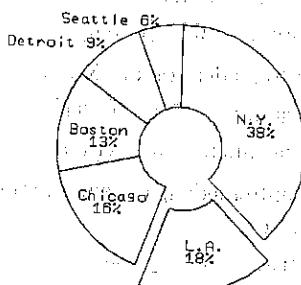
Die Gesamtverkäufe der einzelnen Filialen der BCD Co., Ltd. werden in einem Kreisdiagramm dargestellt. Aus dem Kreisdiagramm wird der Abschnitt für die Filiale in L.A. herausgenommen.

Filiale	N.Y.	L.A.	Chicago	Boston	Detroit	Seattle
Gesamtverkäufe	630	297	270	219	156	98

### ■ AUSDRUCK

(Ausdruck in Farbe. Siehe Seite II).

TTL Amount



*Branch	Sales
N.Y.	630
L.A.	297
Chicago	270
Boston	219
Detroit	156
Seattle	98

## ■ TASTENBETÄTIGUNG

1. **DEF A**

\* \* CIRCLE GRAPH \* \*

Graph title = \_

2. TTL **SPC SML**

A **SML MOUNT ENTER**

No. of data = \_

3. 6 **ENTER**

Item title = \_

4. B **SML RANCH ENTER**

Data title = \_

5. S **SML ALES ENTER**

No. 1

Item name = \_

6. N.Y. **ENTER**

Data = \_

7. 630 **ENTER**

No. 2

Item name = \_

8. L.A. **ENTER**

Data = \_

9. 297 **ENTER**

No. 3

Item name = \_

Gleiche Eingabe wie oben.

10. 98 **ENTER**

Correct? (Y/N) \_

11. **Y ENTER**

Data No. = \_

12. 3 **ENTER**

Data = \_

13. 270 **ENTER**

Correct? (Y/N) \_

14. **N ENTER**

Data sort (Y/N) \_

15. **N ENTER**

Doughnut (Y/N) \_

16. **Y ENTER**

Detaching (Y/N) \_

17. **Y ENTER**

Detaching No. = \_

18. 2 **ENTER**

Draw tbl. (Y/N) \_

19. **Y ENTER**

>

## ■ PROGRAMMLISTING

```

10;"A" CONSOLE 80: USING
20:PAUSE "**CIRCLE GRAPH**"
30:CLEAR : RADIANT
40:GOSUB 870:PRINT "WHAT ARE YOU DOING"
50:U$=CHR$ 27
60:P0=PI /2:P1=0:P2=3*PI /2:P3=PI
70:OPEN "1200:N,B,1,A,L"
80:LPRINT U$;"a"
90:LPRINT U$;"0"
100:LPRINT U$;"bL0"
110:MS=LEN TT$:M$=24
120:IF MS=MJ$XZ THEN 160
130:MJ=18: IF MS$MJ$XZ THEN 160
140:MJ=12: IF MS$MJ$XZ THEN 160
150:MJ=6
160:LPRINT "M";PX-RR$;"";I-30
170:IF MJ=6:LPRINT U$;"?d": GOTO 210
180:IF MJ=12:LPRINT U$;"?b": GOTO 210
190:IF MJ=18:LPRINT U$;"?c": GOTO 210
200:LPRINT U$;"?d"
210:LPRINT U$;"?t"
220:LPRINT U$;"?b"
230:LPRINT "M";PX$;"";PY+30
240:LPRINT "I"
250:FOR K=1 TO YK
260:C(K)= INT (DD(K)/YK*10000+.5)*100
270:NEXT K
280:PP=PI *2/YK:P(0)=0
290:FOR K=1 TO YK
300:P(K)=P(K-1)+PP*DD(K)
310:NEXT K
320:I1= INT (RR*COS P0):I2= INT (RR*SIN P0)
330:01= INT (RR*COS P0):02= INT (RR*SIN P0)
340:Y0=99999:Q=0
350:FOR J=1 TO YK
360:IF Q=4 LET Q=0
370:T1=P0-P(J-1): IF T1<0 LET T1=T1+2*PI
380:T2=P0-P(J): IF T2<0 LET T2=T2+2*PI
390:TM=P0-(P(J-1)-(P(J)-P(J-1))/2)
400:IF TM<0 LET TM=TM+2*PI
410:AX=0:AY=0: IF J>NN THEN 440
420:T=P0-P(J-1)-(P(J)-P(J-1))/2
430:AX= INT (25*COS T):AY= INT (25*SIN T):DX=AX:DY=AY
440:T=P0-P(J): LPRINT CHR$ 27:Q
450:03= INT (RR*COS T):04= INT (RR*SIN T)
460:I3= INT (RR*COS T):I4= INT (RR*SIN T)
470:IF J=1 OR J>NN OR J-1>NN THEN 490
480:GOTO 510
490:LPRINT "M";I1+AX$;"";I2+AY
500:LPRINT "D";I1+AX$;"";I2+AY:GOTO 520
510:LPRINT "M";I1+AX+1$;"";I2+AY+1
520:T=P(J-1): LPRINT U$;Q
530:LPRINT "D"; INT (RR*COS (P0-T))+AY
      ;""; INT (RR*SIN (P0-T))+AY
540:T=T+.05: IF T>=(J) THEN 530
550:LPRINT "D";I3+AX$;"";I4+AY
560:IF J=YK AND J>NN THEN 580
570:IF J=YK THEN 590
580:LPRINT "D";I1+AX$;"";I4+AY
590:IF RD=0 THEN 640
600:T=P(J): LPRINT U$;Q
610:LPRINT "D"; INT (RD*COS (P0-T))+AY
      ;""; INT (RD*SIN (P0-T))+AY
620:T=T-.05: IF T>=(J-1) THEN 610
630:LPRINT "D";I1+AX$;"";I2+AY
640:MJ=12: IF RR<100 LET MJ=6
650:IF MJ=6:LPRINT U$;"?d"
660:IF MJ=12:LPRINT U$;"?b"
670:A= INT ("C(J)*10000+.5")
680:M=LEN Y$(J):A$=STR$ A:N=LEN-A$+2
690:HH=(M+N)*MJ: LPRINT U$;"?b"
700:GOSUB 1630: IF SM=1 THEN 930
710:XM= INT (RR*COS TM)+AY:YM= INT (RR*
      SIN TM)+AY
720:IF ABS (Y0-YM)<MJ+2 THEN 820
730:XL=0:YL=0
740:IF TM>P0 THEN 760
750:XS=XM-HH:YS=YM: GOTO 810
760:IF TM>P3 THEN 780
770:XS=XM-HH:YS=YM-NJ: GOTO 810
780:IF TM>P2 THEN 800
790:XS=XM-HH:YS=YM-NJ: GOTO 810
800:XS=XM:YS=YM-MJ
810:YD=YS: GOTO 900
820:IF TM>P0 THEN 840
830:XS=XM:YS=YD-MJ-2:XL=XS:YL=YS+MJ:
      GOTO 890
840:IF TM>P3 THEN 860
850:XS=XM:YS=YD-MJ-2:XL=XS:YL=YS: GOTO
      890
860:IF TM>P2 LET YS=YD-MJ-2: GOTO 880
870:YS=YD+MJ+2
880:XS=XM:XL=XS:YL=YS+MJ
890:YO=YS: LPRINT U$;"0"
900:LPRINT "M";XS$;"";YS: LPRINT "P";YS
      ;"";A$;"%"
910:IF XL=0 AND YL=0 THEN 930
920:LPRINT "M";XM$;"";YM: LPRINT "D";XL
      ;"";YL
930:01=03:02=04:I1=I3:I2=I4:NL=0
940:Q=Q+1: NEXT J: DEGREE
950:IF F$="Y":GOSUB 1510: END
960:LPRINT "M";-PX;"";-RR-100: END
970:INPUT "Graph title=";TT$: IF LEN TT
      >10 OR TT$="" THEN 970
980:INPUT "No. of data=";YK: IF YK<2 OR
      YK>20 THEN 980
990:DIM DD(YK),Y$(YK),X$(2),P(YK),C(YK)
1000:INPUT "Item title=";IX$(1): IF LEN
      X$(1)>10 OR X$(1)="" THEN 1000
1010:INPUT "Data title=";IX$(2): IF LEN
      X$(2)>10 OR X$(2)="" THEN 1010
1020:FOR L=1 TO YK
1030:PAUSE "No.":STR$ L
1040:INPUT "Item name=";YS(L)
1050:IF LEN Y$(L)>10 OR Y$(L)="" THEN
      1030
1060:INPUT "Data=";DD(L): IF DD(L)=0
      THEN -1060
1070:NEXT L
1080:INPUT "Correct? (Y/N)":AS

```

```

1090:IF A$="N" THEN 1150
1100:IF A$<>"Y" THEN 1080
1110:INPUT "Data No.:";L
1120:IF L<1 OR L>YK THEN 1110
1130:INPUT "Data=";DD(L): IF DD(L)=0
    THEN 1130
1140:GOTO 1080
1150:INPUT "Data sort (Y/N)";A$
1160:IF A$="N" THEN 1320
1170:IF A$<>"Y" THEN 1150
1180:I=1
1190:K=I+1
1200:IF I=YK AND K>YK THEN 1320
1210:IF DD(I)>DD(K) THEN 1240
1220:w=DD(I):DD(I)=DD(K):DD(K)=w
1230:A$=Y$(I):Y$(I)=Y$(K):Y$(K)=A$
1240:K=K+1
1250:I=I+1
1260:IF I>YK THEN 1280
1270:GOTO 1210
1280:I=I+1
1290:IF I=YK AND K>YK THEN 1320
1300:IF I>YK THEN 1320
1310:GOTO 1190
1320:YT=0
1330:FOR L=1 TO YK:YT=YT+DD(L): NEXT L
1340:WX=480:WY=2580
1350:C$="B"
1360:ZX=300:RR=150:AH=12:ZP=ZX
1370:INPUT "Doughnut (Y/N)";E$
1380:IF E$="N" LET RD=0:ND=0: GOTO 141
    0
1390:IF E$<>"Y" THEN 1370
1400:RD= INT (20*2.5)
1410:INPUT "Detaching (Y/N)";A$
1420:IF A$="N" THEN 1460
1430:IF A$<>"Y" THEN 1410
1440:INPUT "Detaching No.:";NN
1450:IF NN1 OR NN>YK THEN 1440
1460:INPUT "Draw tbi. (Y/N)";F$
1470:IF F$="" OR F$="N" THEN 1490
1480:GOTO 1460
1490:PY=- INT (80+ZY/2):PX= INT (WX/2)
1500:RETURN
1510:Y=-RK-50
1520:LPRINT "M";0;"Y
1530:LPRINT "A"
1540:LPRINT USING "|||||";X$(1);
1550:LPRINT USING "||";" ";
1560:LPRINT USING "|||||";X$(2)
1570:FOR K=1 TO YK
1580:LPRINT USING "|||||";Y$(K);
1590:LPRINT USING "||";" ";
1600:LPRINT USING "###";DD(K)
1610:NEXT K
1620:RETURN
1630:R=(RR+RD)/2:X1= INT (R*COS TM):YM
    = INT (R*SIN TM)
1640:X1=YM/TAN T1:X2=YM/TAN T2:XC=SQR
    (RR*RR-YM*YM)
1650:IF T1>P0 THEN 1780
1660:IF T2>P0 LET PT=1: GOTO 1780
1670:IF T2>P3 LET PT=4: GOTO 1780
1680:IF T2<P2 LET PT=3: GOTO 1780
1690:PT=2: GOTO 1780
1700:IF T1<P2 THEN 1740
1710:IF T2>P2 LET PT=5: GOTO 1780
1720:IF T2>P3 LET PT=6: GOTO 1780
1730:P1=7: GOTO 1780
1740:IF T1<P3 THEN 1770
1750:IF T2>P3 LET PT=8: GOTO 1780
1760:PT=9: GOTO 1780
1770:PT=10
1780:IF PT=2 OR PT=3 OR PT=4 OR PT=7
    OR PT=9 THEN 1800
1790:GOTO 1850
1800:LPRINT "M";XM-(MM*MJ)/2-AX;"";YM-
    MJ/2+AY
1810:LPRINT "P";Y$(J)
1820:LPRINT "M";XM-MJ+AX;"";YM-MJ/2+A
    -MJ-3
1830:LPRINT "P";A$;"Z"
1840:SW=1: RETURN
1850:IF PT=1 LET XS=X1:XE=X2: IF X2>XC
    LET XE=XC
1860:IF PT=5 LET XS=X2:XE=X1: IF X1>XC
    LET XE=XC
1870:IF PT=6 LET XS=X2:XE=X1: IF X2<-X
    C LET XS=-XC
1880:IF PT=6 AND X1>XC LET XE=XC
1890:IF PT=8 LET XS=X2:XE=X1: IF X2<-X
    C LET XS=-XC
1900:IF PT=10 LET XS=X1:XE=X2: IF X1<-
    XC LET XS=-XC
1910:XM=XE-XS-2*MJ: IF XM>MMJ LET SWF
    0: RETURN
1920:LPRINT "M";XS+MJ+(XH-MMJ)/2+AX;""
    ;YM+AY
1930:LPRINT "P";Y$(J)
1940:LPRINT "M";XS+XH/2-MJ+AX;"";YM-M
    J-3+AY
1950:LPRINT "P";A$;"Z"
1960:SW=1: RETURN

```

4103 bytes

## ■ SPEICHERINHALT

< Numerische Variable >

A	✓
I	✓
J	Schleifenzähler
K	✓, Schleifenzähler
L	Schleifenzähler, ✓
M	✓
N	✓
Q	✓
R	✓
T	✓
W	✓
Y	✓, Ausdruckposition für Tabelle
C(YK)	einzelne Daten nach Prozentanteilen
DD(YK)	einzelne Daten
P(YK)	Winkelberechnung, Wert der Daten
AH	✓
AX	X-Richtung für herauszunehmenden Abschnitt
AY	Y-Richtung für herauszunehmenden Abschnitt
DX	X-Richtung für herauszunehmenden Abschnitt
DY	Y-Richtung für herauszunehmenden Abschnitt
HH	✓
I1	X-Startkoordinaten des internen Kreises
I2	Y-Startkoordinaten des internen Kreises
I3	✓
I4	✓
MJ	Zeichengröße des Diagramms und Postentitel
MS	Anzahl Zeichen des Titels der Grafik
NN	herauszunehmende Nr.
NO	✓

O1	X-Startkoordinaten des externen Kreises
O2	Y-Startkoordinaten des externen Kreises
O3	✓
O4	✓
PØ	$\pi/2$
P1	Ø
P2	$3 * \pi/2$
P3	$\pi$
PP	Winkelberechnung
PT	✓
PX	X-Richtung der Grundposition des Diagramms
PY	Y-Richtung der Grundposition des Diagramms
RD	Durchmesser des internen Kreises
RR	Durchmesser des externen Kreises
SW	✓
T1	X-Richtung des Winkels des Abschnitts
T2	Y-Richtung des Winkels des Abschnitts
TM	✓
WX	X-Richtung Papiergröße
WY	Y-Richtung Papiergröße
X1	✓
X2	✓
XC	✓
XE	✓
XL	✓
XM	✓
XS	✓
YK	Anzahl der Daten
YL	✓
YM	✓
YO	✓

YS	✓
YT	Summe der Datenwerte
ZX	✓
ZY	✓

< Zeichenvariable >

A\$	✓
C\$	✓
F\$	✓
U\$	ESC (CHR\$27)
X\$(1)	Postentitel
X\$(2)	Datentitel
Y\$(YK)	Postennamen
TT\$	Titel der Grafik

Es ist sehr hilfreich, wenn die in den PC-1450 eingegebenen Daten vom PC-5000 verarbeitet werden können und wenn Daten vom Blasenspeicher oder vom Diskettenlaufwerk des PC-5000 zum PC-1450 ausgelesen werden können.

In diesem Abschnitt stellen wir ein Programm vor, das die Übertragung von Daten vom PC-1450 in den Magnetblasenspeicher ermöglicht, wenn der PC-1450 und der CE-130T an den PC-5000 angeschlossen werden. Außerdem können mit diesem Programm Daten aus dem Magnetblasenspeichers des PC-5000 in den PC-1450 überspielt werden.

### < VORBEREITUNGEN >

1. In der folgenden Tabelle wird das Verbindungskabel für den Anschluß des CE-130T an den PC-5000 gezeigt.

#### « Verbindungskabel-Tabelle »

Signalname	Stift-Nr.	Stift-Nr.	Signalname
SD	2	1	FG
RD	3	2	SD
RS	4	3	RD
CS	5	4	RS
DR	6	5	CS
SG	7	6	DR
CD	8	7	SG
RR	11	8	CD
ER	20	11	RR

CE-130T

Männlicher Stecker

DB-25P oder äquivalentes Teil

PC-5000

Männlicher Stecker

DB-25P oder äquivalentes Teil

2. Für den Anschluß des CE-130T an den PC-5000 und für den Anschluß von Peripheriegeräten an den PC-1450 muß das Kabel der obigen Tabelle entsprechen.

Hinweis) Vor Eingabe des nachstehend beschriebenen Programms, das für die Verwendung der Schnittstelle RS-232C erforderlich ist, in den PC-5000 muß das folgende Verfahren ausgeführt werden.

- Auf dem Bildschirm muß die folgende Anzeige erscheinen

A >

(Abhängig vom zugewiesenen Laufwerk erscheint anstelle von 'A' manchmal C oder G.)

Dann die folgenden Tasten betätigen

MODE RS-232, C

Damit sind die Vorbereitungen vor Eingabe des Programms in den PC-5000 beendet. Danach die Tasten BASIC drücken, um das BASIC-Programm eingeben zu können.

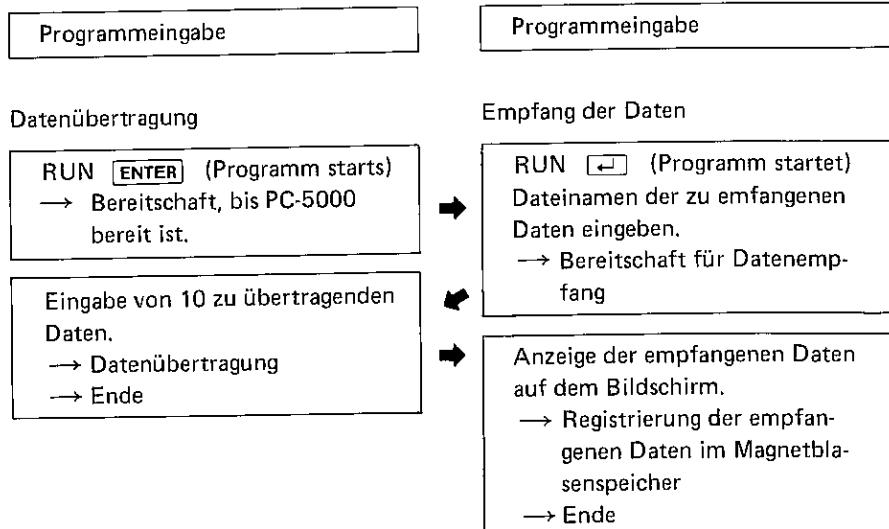
### ★ PC-1450 → PC-5000

Dieses Programm ermöglicht die Eingabe von 10 Daten in den PC-1450 und die Übertragung der Daten zum PC-5000, wo die Daten im Magnetblasenspeicher gespeichert werden. Die übertragenen Daten werden auf dem Bildschirm angezeigt.

### ■ BEDIENFOLGE

PC-1450 (Übertragung)

PC-5000 (Empfang)



## ■ PROGRAMMLISTING

[PC-1450]

```

10:OPEN "1200,N,8,1,A,L"
      "
20:CLEAR :DIM B$(9)
30:FOR C=0 TO 9
40:INPUT B$(C)
50:NEXT C
60:PRINT #1B$(*)"
70:CLOSE
80:END

```

81 bytes

## ■ SPEICHERINHALT

[PC-1450]

C	Schleifenzähler
B\$(9)	Daten

[PC-5000]

C	Schleifenzähler
B\$(9)	Daten
FI\$	Dateiname

## [PC-5000]

```

100 CLEAR:DIM B$(9) AS FILE NAME:FILENAME
20 INPUT "FILE NAME=";FI$
30 OPEN "1200,N,8,1,LF" AS #1
40 FOR C=0 TO 9
50 INPUT #1,B$(C)
60 PRINT B$(C);";"
70 NEXT C
80 OPEN "O",#2,FI$
90 FOR C=0 TO 9
100 PRINT #2,B$(C)
110 NEXT C
120 CLOSE
130 END

```

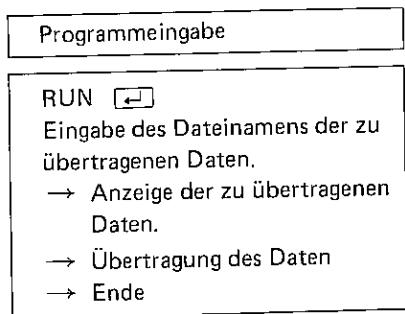
256 bytes

## ★ PC-5000 → PC-1450

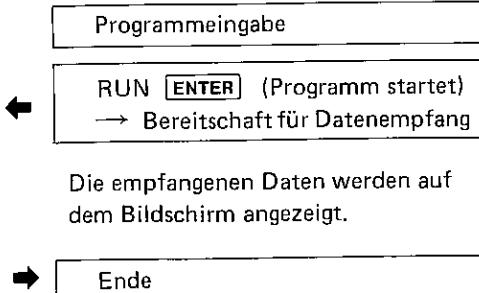
Dieses Programm ermöglicht die Übertragung von Daten, die im Magnetblasenspeicher des PC-5000 gespeichert sind, in den PC-1450. Die übertragenen Daten werden auf dem Bildschirm angezeigt.

## ■ BEDIENFOLGE

PC-5000 (Übertragung)



PC-1450 (Empfang)



## ■ PROGRAMMLISTING

[PC-1450]

```
10:OPEN "1200,N,8,1,A,L"
      "
20:CLEAR :DIM B$(9)
30:INPUT #1B$(*)
40:FOR C=0 TO 9
50:PAUSE B$(C)
60:NEXT C
70:CLOSE
80:END
```

81 bytes

[PC-5000]

```
10 CLEAR:DIM B$(9)
20 INPUT "FILE NAME=":FI$
30 OPEN "COM1:1200.N,8,1,LF" AS #1
40 OPEN "I",#2,FI$
50 FOR C=0 TO 9
60 INPUT #2,B$(C)
70 PRINT B$(C):" ";
80 NEXT C
90 FOR C=0 TO 9
100 PRINT #1,B$(C)
110 NEXT C
120 CLOSE
130 END
```

256 bytes

## ■ SPEICHERINHALT

[PC-1450]

C	Schleifenzähler
B\$(9)	Daten

[PC-5000]

C	Schleifenzähler
B\$(9)	Daten
FI\$	Dateiname

# PROGRAMMTITEL: 3-dimensionale Grafik

Erforderliche Peripheriegeräte  
CE-515P·CE-516L

Computer sind nicht nur zum Arbeiten da, sondern sollen auch Spaß machen. Ein Beispiel dafür ist das Zeichnen von interessanten Grafiken wie mit diesem Programm.

## ■ BEDIENUNG

1. **DEF A** (Programmstart)
2. Entsprechend der Abfrage auf dem Display die Nr. der zu verwendenden Farbe eingeben.
  - 0: schwarz
  - 1: blau
  - 2: grün
  - 3: rot
3. Entsprechend der Abfrage auf dem Display die einzelnen Koordinaten eingeben.

( $Y2 \leq -50$ ,  $Y2 \leq 0$ )

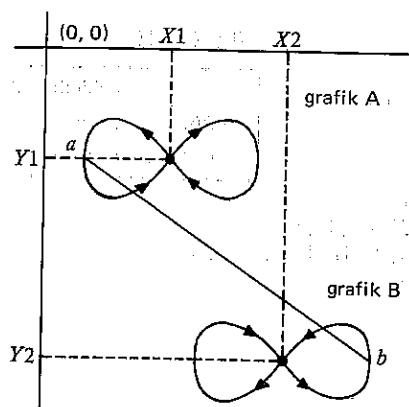
X1      Koordinaten  $x, y$  der Mitte der Grafik  
Y1

X2      maximale Koordinaten  $x, y$  der Grafik  
Y2

Hinweis) X2 muß kleiner gleich 380 und Y2 kleiner gleich 999 sein.

Nach der obigen Eingabe erfolgt der Ausdruck, danach endet das Programm.

## ■ PROGRAMMINHALT



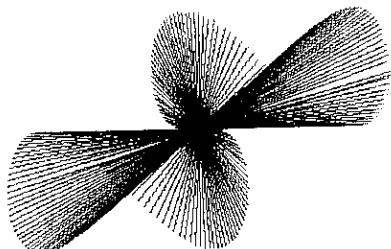
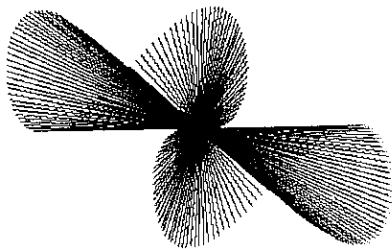
## ■ BEISPIEL

Zeichnen Sie eine Grafik nach den Werten in der folgenden Tabelle.

Nr.	Farbe	X1	Y1	X2	Y2
1	rot	220	-150	100	-50
2	blau	220	0	100	-100

## ■ AUSDRUCK

(Ausdruck in Farbe. Siehe Seite II)



## ■ TASTENBETÄIGUNG

< Beispiel 1 >

1. DEF A

COLOR (0-3) \_

2. 3 ENTER

X1 = \_

3. 220 ENTER

Y1 = \_

Gleiche Eingabe wie oben.

4. -50 ENTER

> \_

< Beispiel 2 >

1. DEF A

COLOR (0-3) \_

2. 1 ENTER

X1 = \_

3. 220 ENTER

Y1 = \_

Gleiche Eingabe wie oben.

4. -100 ENTER

> \_

## ■ PROGRAMMLISTING

```

10: "A",OPEN,"1200,N:8,1,A,L": (CONSOLE,8)
20:CLEAR : WAIT 0: RADIANT:U$=CHR$ 27
30:BEEP 1: INPUT "COLOR (0-3)":C:C=
INT C: IF 0<=C AND C<4 THEN 50
40:GOTO 30
50:BEEP 1: INPUT "X1":;A
60:BEEP 1: INPUT "Y1":;D
70:BEEP 1: INPUT "X2":;B
80:BEEP 1: INPUT "Y2":;E
90:LPRINT U$;"B";U$;STR$ C: LPRINT "M"
00,B": LPRINT "I"
100:FOR I=0 TO 2PI STEP .05:X=COS (I)*
100:Y=XXSIN (I)
110:LPRINT "M";A-X";"10-Y: LPRINT "D";
B+X";"1-E-Y: NEXT I
120:LPRINT U$;"a": BEEP 2: END

```

281 bytes

## ■ SPEICHERINHALT

A	$x_1$
B	$x_2$
C	Farbe
D	$y_1$
E	$y_2$
I	Schleifenzähler
U\$	ESC(CHR\$27)
X	Plottposition auf X-Achse
Y	Plottposition auf Y-Achse

## **PROGRAMMTITEL:    Zahlenspiel**

In diesem Programm werden dreistellige Zufallszahlen erzeugt, die geraten werden müssen.

### **■ BEDIENUNG**

1. **[DEF] [A]** (Programmstart)
2. Auf dem Display wird "X=" angezeigt. Geben Sie eine Zahl ein. Dann wird auf dem Display die Anzahl der bisherigen Versuche und ein Kommentar angezeigt.

(Zeit: 1 Sekunde)

Beispiel:

- Comment: 1    1

In der obigen Anzeige bedeutet die erste 1, daß eine der Stellen der eingegebenen Zahl und die Position dieser Stelle richtig ist. Die zweite 1 bedeutet, daß eine weitere Stelle der eingegebenen Zahl richtig ist, aber die Position dieser Stelle nicht richtig ist.

- Comment: 3    0

Diese Anzeige bedeutet, daß die eingegebene Zahl richtig ist (sowohl die Stellen als auch die Positionen der Stellen). Nach Anzeige einer Meldung und der Anzahl der Versuche endet das Programm.

Hinweis: Die Zahl, die geraten werden muß, hat immer 3 Stellen.

## ■ TASTENBETÄIGUNG

1. **DEF A**

X = \_

2. 123 **ENTER**

1 123

Comment: 0

X = \_

3. 145 **ENTER**

2 145

Comment: 1 0

X = \_

Gleiche Eingabe wie oben.

:

4. 305 **ENTER**

6 305

Comment: 3 0

VERY GOOD ! 6

>

## ■ PROGRAMMLISTING

```

10: "A" CLEAR :RANDOM :Y
      =1
20:FOR A=2 TO 4:A(A)=
      RND 10-1:NEXT A:IF (
      B=C)+(C=D)+(D=B)<>0
      THEN 20
30:BEEP 1:INPUT "X=";X
40:PAUSE Y,X
50:FOR A=6 TO 8:A(A)=X-
      INT ((X*10)*10):X= INT
      (X/10):NEXT A
60:J=0:L=6:P=0:M=0:F=0
70:FOR A=2 TO 4:IF A(A)-
      =A(L) LET J=J+1
80:GOSUB 110:GOSUB 110:
      M=F:F=G:G=H:H=M:L=L-
      2:NEXT A
90:PAUSE "Comment:";
      USING "#####";J;P:IF
      JK>3 LET Y=Y+1:GOTO
      30
100:BEEP 2:PRINT "VERY G
      OOD ! ";Y:END
110:L=L+1:IF A(A)=A(L)
      LET P=P+1
120:RETURN

```

## ■ SPEICHERINHALT

A	✓
B	✓
C	3-stellige Zahl
D	✓
F	✓
G	✓
H	✓
J	Kommentar
L	✓
M	✓
P	Kommentar
X	Eingabewert
Y	Anzahl der Versuche

## ANHANG I. INDEX

ABS . . . . .	181
ACS . . . . .	181
Addition . . . . .	63
AHC . . . . .	181
AHS . . . . .	182
AHT . . . . .	182
AND . . . . .	94
AREAD . . . . .	130
Anzeige . . . . .	24
ASC . . . . .	188
ASN . . . . .	182
ATN . . . . .	182
Auf-/Abrundung . . . . .	56
Ausschalten . . . . .	20
BEEP . . . . .	132
Betriebsart . . . . .	25
Betriebshinweise . . . . .	9
CHR\$ . . . . .	188
CLEAR . . . . .	135
CLOAD . . . . .	116
CLOAD? . . . . .	117
CLOSE . . . . .	191
CONSOLE . . . . .	192
CONT . . . . .	118
COS . . . . .	182
CSAVE . . . . .	119
CUR . . . . .	183
DATA . . . . .	136
Datenspeicherung . . . . .	39
DEGREE/GRAD/RADIAN . . . . .	137, 145, 166
DEG . . . . .	183
DIM . . . . .	138
DMS . . . . .	183
Definable Keys . . . . .	21, 229
Division . . . . .	63
Drucker . . . . .	35
END . . . . .	140
Editierfunktionen . . . . .	46, 85
Einfache Variable . . . . .	98

Einschalten . . . . .	Anh. A, Anh. B	20
EXP . . . . .		183
Exponentenfunktionen . . . . .		70
FACT . . . . .		183
Fakultät . . . . .		70
Fehlermeldung/Fehlersuche . . . . .		112, 216
Feldvariable . . . . .		101
FOR . . . TO . . . STEP . . . . .		141
GOSUB . . . . .		143
GOTO . . . . .		120, 144
HCS . . . . .		184
Hexadezimalumwandlung . . . . .		59
Hexadezimalzahlen . . . . .		59
HSN . . . . .		184
HTN . . . . .		184
Hyperbel- und inverse Hyperbelfunktionen . . . . .		69
IF . . . THEN . . . . .		146
Indizierte Variable . . . . .		98
INKEY\$ . . . . .		178
INPUT . . . . .		147
INPUT# . . . . .		149
INPUT#1 . . . . .		193
INT . . . . .		184
Kassetten-Interface . . . . .		38
Koordinatenumwandlung . . . . .		59
LEFT\$ . . . . .		189
LEN . . . . .		189
LET . . . . .		152
LIST . . . . .		121
LLIST . . . . .		122, 195
LN . . . . .		184
LOAD! . . . . .		197
LOG . . . . .		184
LPRINT . . . . .		153, 155, 199
Logarithmische Funktionen . . . . .		70
Löschen einer Zeile . . . . .		111
Löschen der Eingabe oder einer Fehlermeldung . . . . .		55
MEM . . . . .		179
MID\$ . . . . .		189
Multiplikation . . . . .		63
NEW . . . . .		127

NEXT . . . . .	157
Nachkommastellen . . . . .	56
NOT . . . . .	94
Numerische Feldvariable . . . . .	101
ON . . . GOSUB . . . . .	158
ON . . . GOTO . . . . .	159
OPEN . . . . .	202
OPEN\$ . . . . .	204
OR . . . . .	94
PASS . . . . .	128
PAUSE . . . . .	160
PI . . . . .	180
PRINT . . . . .	162
PRINT# . . . . .	164
PRINT#1 . . . . .	205
Programmaufbau . . . . .	108
Programmausführung . . . . .	111
Programmeingabe . . . . .	108
Programmerstellung . . . . .	107
POL . . . . .	185
Potenzfunktion . . . . .	64
RANDOM . . . . .	167
RCP . . . . .	185
READ . . . . .	168
Rechenbereich . . . . .	77
Rechengenauigkeit . . . . .	83
REM . . . . .	169
RESTORE . . . . .	170
RETURN . . . . .	171
Reziprok-Rechnung . . . . .	65
RIGHT\$ . . . . .	189
RND . . . . .	185
ROT . . . . .	186
RUN/GOTO . . . . .	129
SAVE . . . . .	207
Schlüsselwörter, vorprogrammierte . . . . .	85, 229
SGN . . . . .	186
SIN . . . . .	186
SQR . . . . .	187
SQU . . . . .	187
Speicherrechnung . . . . .	65

Statistische Berechnungen . . . . .	71
STOP . . . . .	172
STR\$ . . . . .	189
Stromversorgung . . . . .	11
Subtraktion . . . . .	63
TAN . . . . .	187
Tastaturabdeckung . . . . .	17
Tastenfunktionen . . . . .	228
TEN . . . . .	187
TEXT . . . . .	209
Textausdrücke . . . . .	93
Textfunktionen . . . . .	188
Textfeldvariable . . . . .	102
Trigonometrische Funktionen und Umkehrfunktion . . . . .	69
TRON . . . . .	174
TROFF . . . . .	173
USING . . . . .	175
Vergleichsausdrücke, logische . . . . .	89, 94
VAL . . . . .	190
Variable A . . . . .	99
Variablen . . . . .	96
WAIT . . . . .	177
Winkelumwandlung . . . . .	58
Wissenschaftliche Schreibweise . . . . .	57, 84
Wurzelfunktion . . . . .	64
A . . . . .	1
A\$ . . . . .	203
B . . . . .	92, 101
C . . . . .	101, 109
D . . . . .	101, 109
E . . . . .	101, 109
F . . . . .	101, 109
G . . . . .	101, 109
H . . . . .	101, 109
I . . . . .	101, 109
L . . . . .	101, 109
M . . . . .	101, 109
N . . . . .	101, 109
O . . . . .	101, 109
P . . . . .	101, 109
R . . . . .	101, 109
S . . . . .	101, 109
T . . . . .	101, 109
U . . . . .	101, 109
V . . . . .	101, 109
W . . . . .	101, 109
X . . . . .	101, 109
Y . . . . .	101, 109
Z . . . . .	101, 109



**SHARP CORPORATION**  
**Osaka, Japan**

1985 © SHARP CORPORATION  
Printed in Japan/Imprimé au Japon  
5E3,8T(TINSG4434CCZ)