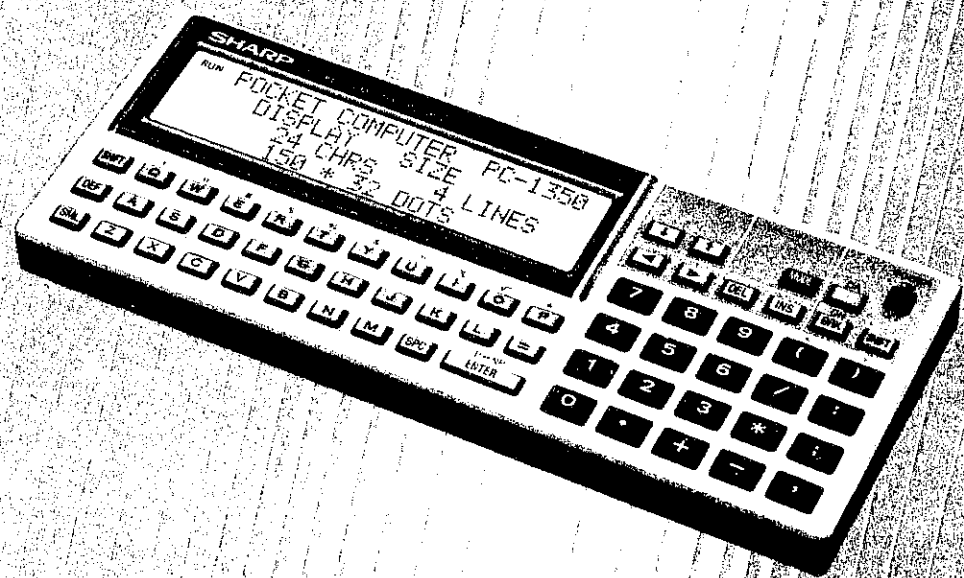


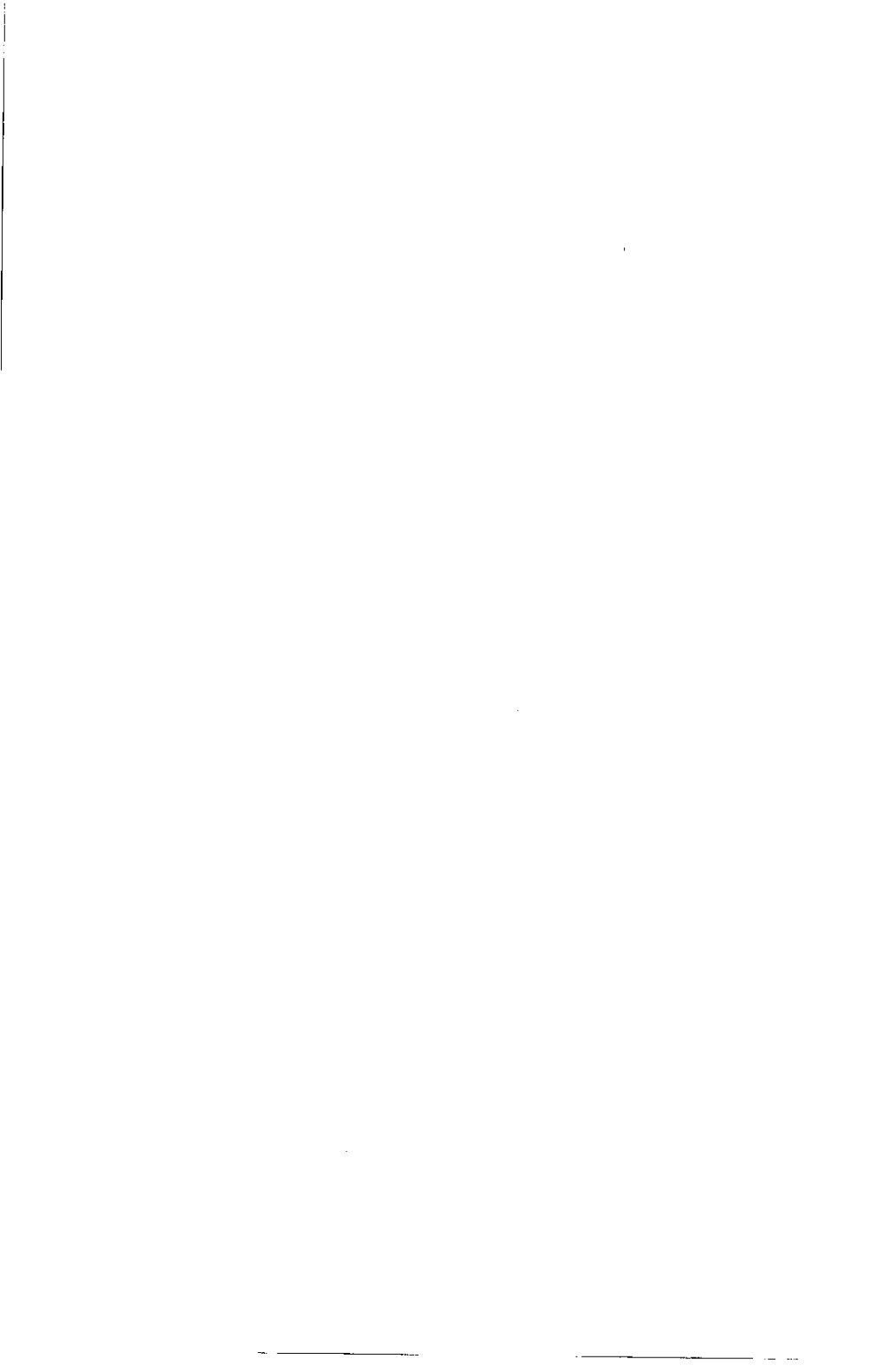
# SHARP®

## WASCHENCOMPUTER

MODELL **PC-1350**

### BEDIENUNGSANLEITUNG





## INHALTSVERZEICHNIS

EINLEITUNG		E-1
KAPITEL 1	WIE MAN DIESES HANDBUCH BENUTZT	1-1
KAPITEL 2	EINFÜHRUNG IN DEN PC 1350	2-1
KAPITEL 3	RECHNEN OHNE PROGRAMMUNTERSTÜTZUNG	3-1
KAPITEL 4	BEGRIFFE UND AUSDRÜCKE DES BASIC	4-1
KAPITEL 5	PROGRAMMIEREN MIT DEM PC 1350	5-1
KAPITEL 6	ABKÜRZUNGEN	6-1
KAPITEL 7	DER GEBRAUCH DES CE-126P DRUCKER/ CASSETTENRECORDER-INTERFACE	7-1
KAPITEL 8	DIE RAM-KARTE (optional)	8-1
KAPITEL 9	BASIC-REFERENZ-TEIL	9-1
	Kommandos	9-4
	Befehle	9-19
	Funktionen	9-70
	Pseudeovariable	9-70
	Numerische Funktionen	9-76
	String-Funktionen	9-80
	Graphik-Kommandos	9-82
	Ein/Ausgabe-Kommandos	9-93
	Text-Funktions-Kommandos	9-107
KAPITEL 10	PANNENHILFE	10-1
	Bedienung des Geräts	10-1
	Fehlersuche in BASIC	10-2

KAPITEL 11	DIE INSTANDHALTUNG DES PC 1350	11-1
	PROGRAMMBEISPIELE	P-1
ANHANG A	FEHLERMELDUNGEN	A-1
ANHANG B	ASCII-CODE TABELLE	B-1
ANHANG C	FORMATIEREN DER DATENAUSGABE	C-1
ANHANG D	BEWERTUNG VON AUSDRÜCKEN UND OPERATOREN-VORRANG	D-1
ANHANG E	TASTENFUNKTIONEN	E-1
ANHANG F	SYMBOLE DES E/A ANSCHLUSSES	F-1
ANHANG G	TECHNISCHE DATEN	G-1
ANHANG H	DIE BENUTZUNG VON PROGRAMMEN, DIE FÜR DEN PC 1245, PC 1250/51 ODER DEN PC 1401 ENTWICKELT WURDEN	H-1
	MODIFIKATIONEN VON PROGRAMMEN DER PC 1245 SERIE	H-1
	ÄNDERUNGEN BEZÜGLICH DES PC 1401	H-3
	ÄNDERUNGEN BEZÜGLICH DES PC 1260/1261	H-4
	ÄNDERUNGEN BEZÜGLICH DER PC 1210 SERIE	H-5

## EINLEITUNG

## WILLKOMMEN IN DER WELT DER SHARP-BESITZER!

Nur wenige Industriezweige der Welt können heute Schritt halten mit dem schnellen Wachstum und dem technischen Fortschritt im Bereich der Personalcomputer. Computer, die noch vor kurzer Zeit einen ganzen Saal gefüllt hätten, deren Programmierung einen akademischen Abschluß erfordert und die Tausende von Dollars gekostet hätten, passen heute in Ihre Handfläche, sind einfach zu programmieren und kosten so wenig, daß fast jeder sie sich leisten kann.

Ihr neuer SHARP PC 1350 wurde entwickelt, um Ihnen den allerneuesten Stand dieser Computerrevolution zugänglich zu machen. Als einer der raffiniertesten Taschencomputer unserer Zeit verfügt er über eine Vielzahl fortschrittlichster Fähigkeiten:

- **SPEICHERSCHUTZ** - der PC 1350 "merkt sich" gespeicherte Programme und Variablen, selbst wenn Sie ihn abschalten.
- **Stromversorgung durch Batterien:** ideal für alle, die ein wirklich portables System benötigen.
- **AUTOMATISCHE ABSCHALTUNG**, die die Batterien schützt, indem die Stromversorgung abgeschaltet wird, wenn innerhalb eines bestimmten Zeitraums keine Operation durchgeführt wird.
- Eine erweiterte Version des BASIC, die formatierte Datenausgabe ermöglicht, weiterhin zweidimensionale Felder, variable String-Länge, Programmverkettung und viele andere fortschrittliche Arbeitsmöglichkeiten.
- **Als Option:** Mit dem Drucker/Mikrocassetten-Interface (Modell CE-126P) können Sie dauerhafte Kopien von Programmen und Daten erstellen. Das Cassetten-Interface erlaubt es Ihnen, einen Cassettenrecorder anzuschließen, um Programme und Daten auf Cassette zu speichern.

Wir gratulieren Ihnen zu Ihrem Eintritt in eine aufregende und faszinierende neue Welt. Wir sind sicher, daß Sie Ihre neue Errungenschaft schon bald als eine der klügsten betrachten werden, die Sie je getätigt haben. Der SHARP PC 1350 ist ein hilfreiches Werkzeug, das entwickelt wurde, um Ihren spezifischen mathematischen, wissenschaftlichen, technischen, kaufmännischen und persönlichen Computerbedürfnissen gerecht zu werden. Mit dem SHARP PC 1350 können Sie JETZT damit anfangen, die Lösungen zu finden, die Sie morgen brauchen werden.

## KAPITEL 1

## WIE MAN DIESES HANDBUCH BENUTZT

Dieses Handbuch soll Sie einführen in die Fähigkeiten und Charakteristika Ihres PC 1350 und es soll Ihnen als wertvolles Nachschlagewerk dienen. Ob Sie ein "Einsteiger" sind oder ob die Computer zu Ihren "alten Bekannten" zählen: Sie sollten sich mit dem PC 1350 vertraut machen, indem Sie die Kapitel 2 bis 6 lesen und durcharbeiten.

- Kapitel 2 beschreibt das äußere Erscheinungsbild Ihres PC 1350.
- Kapitel 3 erklärt den Gebrauch des PC 1350 als normalen Taschenrechner.
- Kapitel 4 erklärte einige Ausdrücke und Begriffe, die für die Programmierung in BASIC von Bedeutung sind, und es erläutert Ihnen die spezielle Bedeutung dieser Begriffe für die Arbeit mit dem PC 1350.
- Kapitel 5 dient als Einführung in die BASIC-Programmierung auf dem PC 1350. Es zeigt, wie man Programme eingibt, korrigiert und abarbeiten läßt.
- Kapitel 6 schließlich behandelt einige Tips und Kniffe, mit denen Sie sich die Arbeit mit Ihrem PC 1350 noch weiter vereinfachen und noch mehr Spaß damit haben können.

Erfahrene BASIC-Programmierer können dann noch Kapitel 9 durchlesen, um die spezifischen Züge des BASIC im PC 1350 kennenzulernen. Da sich alle BASIC-Dialekte geringfügig unterscheiden, empfiehlt es sich, dieses Kapitel mindestens einmal durchzulesen, bevor man mit dem ernsthaften Programmieren beginnt.

Kapitel 9 ist ein Nachschlageteil, der alle Befehle, Kommandos und Funktionen des BASIC in übersichtlicher, alphabetisch gruppierter Anordnung enthält.

Wenn Sie bislang noch nicht in BASIC programmiert haben, schlagen wir vor, daß Sie sich hierfür ein separates Lehrbuch zulegen oder an einem BASIC-Kurs teilnehmen, ehe Sie diese Kapitel durcharbeiten. Dieses Handbuch ist nicht als Lehrbuch für das Programmieren konzipiert.

Der übrige Teil des Handbuchs besteht aus:

- Kapitel 7 - Grundlegende Informationen über die Option CE-126P, Drucker/Mikrocassetten-Interface.
- Kapitel 8 - Erklärungen zum Gebrauch der Option CE-201M/202M RAM.
- Kapitel 10 - Ein Leitfaden, der Ihnen helfen soll, Probleme der Bedienung und Programmierung zu lösen.
- Kapitel 11 - Pflege und Wartung Ihres neuen Computers.

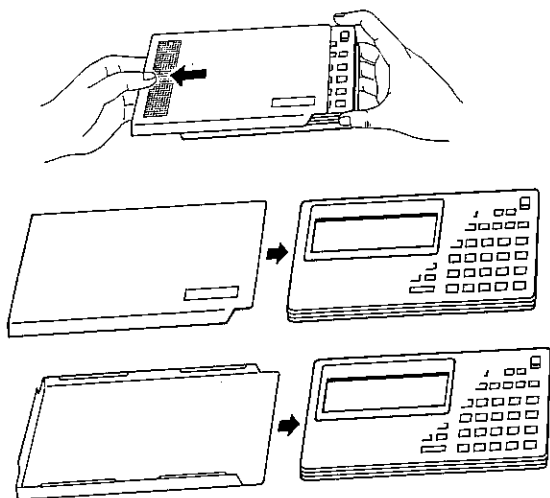
Ein ausführlicher Anhang am Ende dieses Handbuches stellt Ihnen nützliche Tabellen, Vergleichsübersichten und einige weitere Hinweise zum Gebrauch und zur Funktionsweise Ihres PC 1350 zur Verfügung.

#### Über den Gebrauch der Abdeckung

Wenn Sie Ihren Computer nicht benutzen, sollten Sie die Eingabetastatur mit der dafür vorgesehenen Schutzhülle abdecken.

\* Wenn der Computer in Betrieb ist:

Entfernen Sie die Abdeckung, so wie es in der untenstehenden Abbildung gezeigt ist.

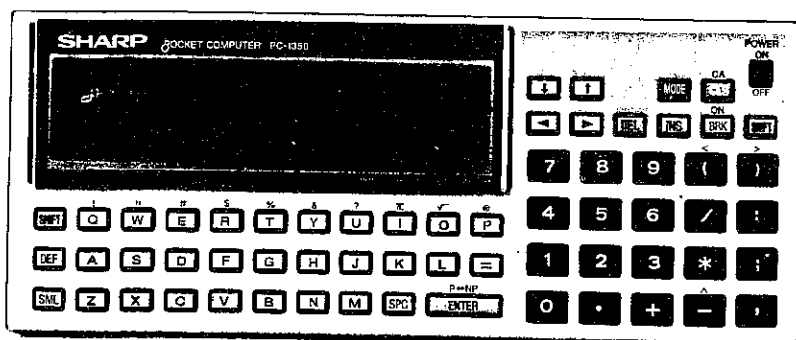


## KAPITEL 2

### EINFÜHRUNG IN DEN PC1350

Das System SHARP PC 1350 besteht aus:

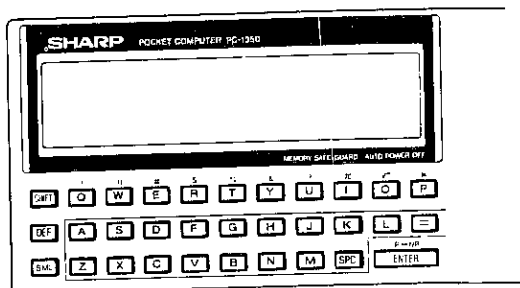
- 62-Zeichen-Tastatur
- 4-zeilige 24-Zeichen-Anzeige oder Grafik-Display mit 32 x 150 Punkten
- 8-Bit CMOS Prozessor
- 40 KB ROM
- 5 KB RAM (Standard)
- Option: CE-126P (Drucker/Mikrocassetten-Interface)



Um Sie mit der Anordnung und den Funktionen von Teilen der Tastatur Ihres PC 1350 vertraut zu machen, wollen wir nun die einzelnen Bestandteile der Tastatur genauer betrachten. Für den Anfang sollten Sie zunächst die Positionen der einzelnen Tasten feststellen und die dazugehörigen Beschreibungen lesen. In Kapitel 3 werden wir anfangen, mit unserem neuen Gerät zu arbeiten.



## BESCHREIBUNG DER TASTATUR



**A** - **Z**

Alphabetische Tastatur. Sie kennen diese wahrscheinlich von einer normalen Schreibmaschine. Wenn Sie diese Tasten betätigen, werden (im Gegensatz zu den meisten Schreibmaschinen) Großbuchstaben angezeigt - das ist jedoch unbedingte Voraussetzung für den Umgang mit dem PC 1350, da dieser nur Kommandos in Großbuchstaben versteht. Wenn Sie die Taste **SHIFT** zuerst drücken und dann den gewünschten Buchstaben, werden Kleinbuchstaben angezeigt.

**=**

Gleichheitszeichen. Auf dem PC 1350 wird diese Taste nicht benutzt, um das Ende einer Rechenoperation anzuzeigen: dieses Symbol hat eine spezielle Funktion in der BASIC-Programmierung.

**SPC**

Leertaste. Nach Betätigen dieser Taste rückt der Cursor um einen Schritt nach rechts, wobei ein Leerfeld hinterlassen wird. Betätigt man **SPC**, während der Cursor auf einem Buchstaben steht, wird dieser Buchstabe mit einem Leerfeld überschrieben.

**P=NP**  
**ENTER**

**ENTER**-Taste. Wenn man diese Taste betätigt, wird, was immer vorher eingegeben wurde, in den Speicher des Computers befördert. Die Funktion ähnelt u. a. der Wagenrücklauf-taste auf einer Schreibmaschine. Man muß **ENTER** drücken, bevor der PC 1350 auf alphanumerische Eingaben über die Tastatur reagiert. Drückt man **SHIFT** vor Betätigung der **ENTER**-Taste, bewirkt das, daß manuelle Rechenoperationen auf dem CE-126P-Drucker ausgedruckt werden.

**DEF**

Diese spezielle Taste wird zur Abarbeitung von BASIC-Programmen benutzt.

**SHIFT**

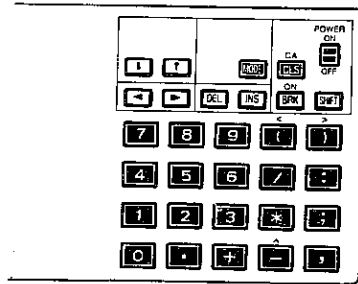
Doppelfunktionstaste. Erstens: Wenn Sie diese Taste vor einer anderen Taste betätigen, wird das über der Taste stehende Zeichen angezeigt. Zweitens: Wenn Sie **SHIFT** vor der Taste **MODE** betätigen, wird der **RESERVE-Modus** aktiviert. Normalerweise wird mit der **MODE**-Taste der **RUN-** oder **PROGRAMM-Modus** aufgerufen. Durch die Benutzung der Taste **SHIFT**, ergibt sich nun eine dritte Möglichkeit (Lesen Sie dazu auch die entsprechenden **MODUS-Abschnitte** in diesem Kapitel!). Drittens: Wenn Sie die Taste **SHIFT** vor einer alphabetischen Taste benutzen (alle alphabetischen Tasten sind normalerweise Großbuchstaben), wird ein Kleinbuchstabe angezeigt. Viertens: Wenn Sie die Taste **SHIFT** vor einer alphabetischen Taste drücken und gleichzeitig durch die **SML**-Taste die Kleinschreibungsfunktion aktiviert ist, werden Großbuchstaben angezeigt.

**SML**

Die Taste **SML** wird für die Ausgabe von Kleinbuchstaben benutzt. Wenn Sie die Taste **SML** erneut drücken, werden wieder Großbuchstaben angezeigt. Beachten Sie bitte, daß Sie im Kleinschreibungsmodus durch Drücken der Taste **SHIFT** vor einer anderen Taste lediglich die Ausgabe eines einzigen Großbuchstabens bewirken können.

- ! " #
- \$ % &
- ? π √
- ⓐ

Diese Symbole befinden sich über der oberen Reihe der Alpha-Tastatur. Betätigt man **SHIFT** und dann den Buchstaben unter dem entsprechenden Zeichen, erscheinen diese Symbole in der Anzeige.



POWER

ON



OFF

Benutzen Sie diese Taste, um den PC 1350 ein- bzw. auszuschalten.

**SHIFT**

**SHIFT**

-Taste. Siehe oben.

CA  
CLS

Die rote **CLS**-Taste erlaubt es Ihnen, den gesamten Bildschirminhalt zu löschen. Wenn Sie **SHIFT** vor **CLS** drücken, aktivieren Sie die zweite CLEAR-ALL-Funktion. Dadurch wird der Computer initialisiert. Beachten Sie bitte, daß dadurch die Programme, die Reservespeicher und die Variablen nicht gelöscht werden. Um alle Programme zu löschen, geben Sie **NEW** ein und drücken **ENTER**, während Sie sich im PROGRAMM-Modus befinden. Um alle Reservespeicher zu löschen, geben Sie **NEW** ein und drücken **ENTER**, während Sie sich im RESERVE-Modus befinden. Um alle Variablen auf Null zu setzen, drücken Sie **CLS** und anschließend **ENTER**.

MODE

Wenn Sie den PC 1350 einschalten, achten Sie darauf, welchen Modus Sie gewählt haben (RUN, PROGRAMM oder RESERVE). Drücken Sie die Taste **MODE**, um den jeweiligen Modus zu ändern (z. B. von RUN zu PROGRAMM oder von PROGRAMM zu RUN). Drücken Sie **SHIFT** und dann **MODE**, um den RESERVE-Modus auszuwählen. Um den RESERVE-Modus zu verlassen, drücken Sie **MODE**. Alle Berechnungen (wenn Sie den Computer als Rechner benutzen) werden im RUN-Modus ausgeführt und alle Programme laufen im RUN-Modus. Alle Programme werden im PROGRAMM-Modus geschrieben und ausgegeben. Der RESERVE-Modus wird benutzt, um häufig benutzte Funktionen auf einzelnen Tasten zu speichern; bei gespeicherten Menüs wird er dazu benutzt, diese Funktionen schnell zu identifizieren.

ON  
BRK

**BREAK** -Taste. Die **BRK**-Taste wird zur Unterbrechung des Programmlaufs benutzt. Nach Abschalten des Rechners durch die Automatik wird er mit dieser Taste wieder eingeschaltet.

INS

Einfügen-Taste. Diese Taste ermöglicht die Bewegung des Cursors nach rechts, ohne daß vorher eingegebene Zeichen gelöscht werden. Im **SHIFT**-Modus setzt diese Taste einen Leerschritt unmittelbar vor das Zeichen, auf dem der Cursor jeweils steht, so daß Sie hier eine Einfügung machen können.

DEL

Lösch-Taste. Diese Taste ermöglicht das Löschen jedes Zeichens, auf dem der Cursor gerade steht.

↑

Pfeil-nach-oben-Taste. Diese Taste bewirkt, daß die vorherige Zeile ins Display gerufen wird.

↓

Pfeil-nach-unten-Taste. Diese Taste bewirkt, daß die nachstehende Zeile angezeigt wird.

- ◀ Rückschritt-Taste. Mit Hilfe dieser Taste können Sie den Cursor nach links bewegen, ohne dabei vorher eingegebene Zeichen zu löschen.
- ▶ Vorwärts-Taste. Mit Hilfe dieser Taste können Sie den Cusor nach rechts bewegen, ohne dabei vorher eingegebene Zeichen zu löschen.
- 0 - 9 . Zifferntasten. Die Anordnung dieser Tasten ähnelt der auf einem gewöhnlichen Taschenrechner.
- / Divisionszeichen. Mit dem Schrägstrich gibt man dem PC 1350 zu verstehen, daß man eine Division ausführen möchte.
- \* Multiplikationszeichen. Der Stern steht beim PC 1350 wie bei den meisten anderen Computern für die Multiplikation.
- ^ Subtraktionszeichen. Mit diesem Zeichen geben Sie Subtraktionen ein. Im **SHIFT** -Modus fungiert es als Exponentiationszeichen, das dem Computer anzeigt, daß eine Zahl potenziert werden soll.
- + Additionszeichen.
- < Klammerzeichen links. Wenn Sie **SHIFT** drücken, wird die Relation "kleiner als" hergestellt.
- > Klammerzeichen rechts. Wenn Sie **SHIFT** drücken, wird die Relation "größer als" hergestellt.
- : Doppelpunkt-Zeichen.
- ; Semikolon-Zeichen.
- ,

BESCHREIBUNG DES ZEILENDISPLAYS

```

RUN
PRO

SML
SHIFT
DEF
```

Der PC 1350 hat eine programmierbare 32 x 150 Punkt-Flüssigkristall-anzeige. Das Display kann bis zu 24 Zeichen pro Zeile und insgesamt 4 Zeilen anzeigen. Jedes Zeichen braucht zur seiner Darstellung 5 x 7 Punkte.

ei graphischen Darstellungen kann das gesamte Display als 32 x 150 Punkt-Matrix benutzt werden. Einzelne Punkte können innerhalb der 150 Spalten so besetzt werden, daß man mit ihrer Hilfe Graphiken, beliebige Figuren und spezielle Symbole erzeugen kann.

Das Anzeigefeld besteht aus:

> Bereitschaftssymbol. Es erscheint, wenn der Computer bereit ist, eine Eingabe aufzunehmen. Sobald Sie mit der Eingabe beginnen, verschwindet das Bereitschaftssymbol und wird durch den Cursor ersetzt.

Cursor. Dieses Symbol (Unterstrich) markiert die Stelle des nächsten einzugebenden Zeichens. Wenn man mit der Eingabe von Daten beginnt, ersetzt der Cursor das Bereitschaftssymbol. Als Markierungssymbol wird der Cursor auch im Zusammenhang mit den INSERT- und DELETE-Funktionen benutzt.

RUN RUN-Symbol. Zeigt den RUN-Modus für den PC 1350 an.

PRO PROGRAMM-Symbol. Zeigt den PROGRAMM-Modus für den PC 1350 an.

**Beachten Sie:** Wenn im Display weder RUN noch PRO erscheinen, befindet sich der PC 1350 im RESERVE-Modus (RSV).

SML Anzeige für Kleinschriftmodus. Wenn im Display "SML" erscheint, werden bei Betätigung der Alphatastatur Kleinbuchstaben angezeigt. Der Kleinschriftmodus wird durch erneutes Betätigen von SML aufgehoben, der Computer kehrt in den Großschriftmodus zurück.

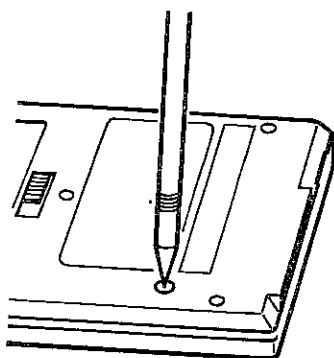
SHIFT Anzeige der Doppelfunktion. Der Schriftzug leuchtet auf, wenn die Doppelfunktionstaste betätigt wurde. Bitte beachten Sie, daß die SHIFT-Taste wieder losgelassen werden muß, bevor eine andere Taste betätigt wird.

DEF Define-Modus. Diese Anzeige leuchtet auf, wenn die DEF-Taste betätigt wurde.

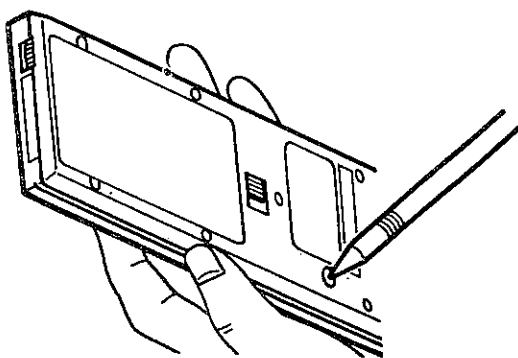
Bei Eingaben, die länger sind als 4 Zeilen:

Das Display des PC 1350 besteht aus 4 Zeilen (je 24 Zeichen pro Zeile). Alle Eingaben oder Rechenergebnisse werden jeweils vom oberen Rand aus angezeigt. Falls die Eingaben oder Ergebnisse das Maximum von 4 Zeilen überschreiten, wird der Bildschirminhalt einfach jeweils 1 Zeile nach oben verschoben (d. h. die erste Zeile der Anzeige rollt nach oben und ist somit auf dem Display nicht mehr zu sehen).

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 PC 1350 zu initialisieren, drücken Sie irgendeine Taste, halten diese fest und drücken gleichzeitig den ALL-RESET-Schalter auf der Rückseite des Gerätes. Die Programme, Variablen und der Reservespeicher werden dabei nicht gelöscht.



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 PC 1350 nach dieser Operation immer noch nicht reagieren, betätigen Sie den ALL-RESET-Schalter, ohne dabei eine andere Taste zu drücken. Dabei werden allerdings Speicherinhalte, 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.

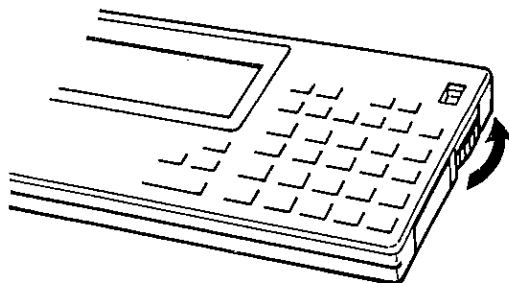
### FUNKTIONEN DES ALL-RESET-SCHALTERS

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 verstärkt, in entgegengesetzter Richtung verringert.

### BATTERIEWECHSEL

Der PC 1350 arbeitet ausschließlich mit Lithium-Batterien. Wenn er an die Option CE-126P angeschlossen ist, kann der PC 1350 auch über diese versorgt werden, sofern deren momentane Betriebsspannung höher ist. Dies verringert den Stromverbrauch der Lithium-Zellen.

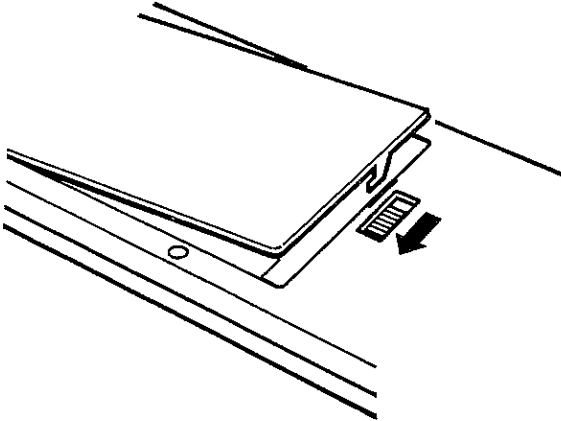
beim Wechseln der Batterien beachten Sie bitte unbedingt die folgenden Hinweise:

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

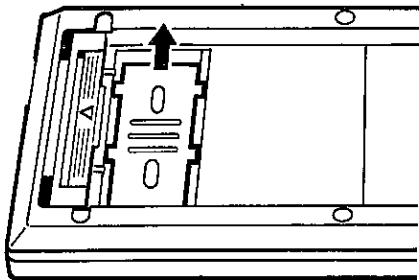
## EINSETZEN DER BATTERIEN

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

- (1) Schalten Sie den Computer ab, indem Sie den Schiebeschalter in die OFF-Position bringen.
- (2) Entfernen Sie den rückwärtigen Gehäusedeckel, indem Sie den Verriegelungsknopf (siehe Abbildung 1) in Pfeilrichtung schieben.

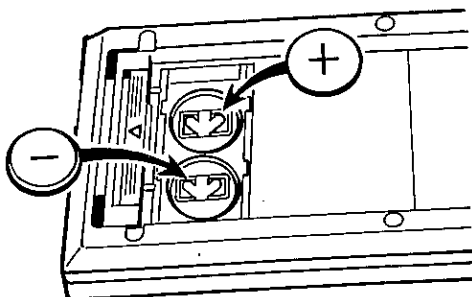


- (3) Wenn Sie einen zusätzlichen RAM-Speicher benutzen, entfernen Sie diesen wie in der KAPITEL 8 beschrieben.
- (4) Schieben Sie die Batterie-Abdeckung in Pfeilrichtung, sie läßt sich dann abheben.

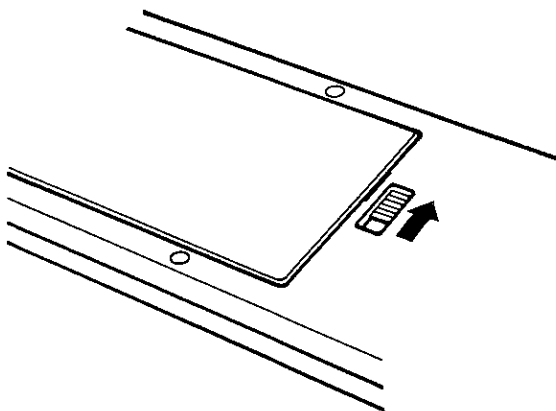




- (5) Erneuern Sie beide Batterien.



- (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.

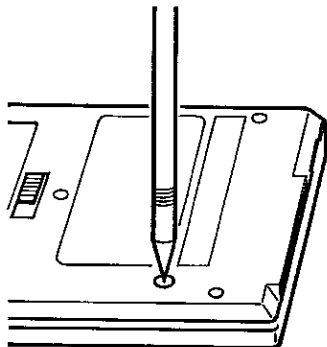


Beachten Sie: Vergewissern Sie sich, daß der Riegel in der LOCK-Position ist: Der PC 1350 arbeitet sonst nicht. Wenn Sie den Computer einschalten, ohne daß der Riegel in der LOCK-Position ist, schieben Sie den Riegel in die LOCK-Position, schalten den Computer wieder aus und anschließend wieder ein. Vergewissern Sie sich, daß die RAM-karte richtig eingesetzt ist.

## ÜBERPRÜFUNG DER FUNKTIONSFÄHIGKEIT NACH DEM BATTERIEWECHSEL

Führen Sie nach dem Batteriewechsel folgende Operationen aus:

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



- (2) Geben Sie folgendes ein:


CALL &EF88

POKE &6EAF,0

In der Anzeige sollte stehen:

```

RUN MODE
CALL &EF88
POKE &6EAF,0
  
```

 Bereitschaftssymbol

Sollte die Anzeige nicht wie oben dargestellt aussehen, wiederholen Sie die Schritte (1) und (2).

CALL &EF88 initialisiert das serielle E/A-Interface.  
POKE &6EAF,0 initialisiert die Anzeige-Funktionen.

Führen Sie diese Initialisierungen nicht durch, so kann es sein, daß Sie falsche Anzeigen oder Fehlfunktionen bei Geräten, die über das serielle E/A-Interface verbunden sind, erhalten.

Wenn die Anzeige leer ist oder ein anderes Zeichen als das Bereitschaftssymbol erscheint, entfernen Sie die Batterien und setzen Sie sie erneut ein, kontrollieren Sie dann wieder das Display.

- (3) Wenn Sie den zusätzlichen RAM-Speicher entfernt haben, setzen Sie ihn jetzt wieder ein. Dazu lesen Sie bitte KAPITEL 8. Denken Sie daran, daß Sie zuvor die Position der Batterien nochmals überprüfen, ehe Sie den RAM-Speicher einsetzen.

<b>Achtung:</b>	Läßt man eine leere Batterie im Computer, kann er durch Säureaustritt aus der Batterie geschädigt werden. Leere Batterien umgehend erneuern!
-----------------	--

<b>Vorsicht:</b>	Bewahren Sie die Batterien außerhalb der Reichweite von Kindern auf!
------------------	--

## KAPITEL 3

## RECHNEN OHNE PROGRAMMUNTERSTÜTZUNG

Nachdem Sie nun vertraut sind mit den Bestandteilen Ihres SHARP PC 1350 und deren Anordnung, wollen wir damit beginnen, die faszinierenden Fähigkeiten Ihres neuen Computers zu entdecken.

Weil der PC 1350 über alle gebräuchlichen mathematischen Funktionen verfügt und darüber hinaus in BASIC programmierbar ist (nützlich für komplexere Rechenoperationen), wird er häufig als "schlauer" Rechner bezeichnet. Das macht natürlich aus Ihnen einen "schlauem" Benutzer.

(Bevor Sie nun mit Ihrem PC 1350 arbeiten, vergewissern Sie sich, daß die Batterien korrekt eingesetzt sind.)

## EINSCHALTEN DES COMPUTERS

Um den PC 1350 einzuschalten, betätigen Sie den Schiebeschalter. Um den PC 1350 wie einen einfachen Taschenrechner zu benutzen, muß der RUN-Modus gewählt werden. Benutzen Sie dazu die **MODE**-Taste, bis "RUN" an der linken Seite des Anzeigefeldes erscheint. Wenn das Gerät eingeschaltet ist, erscheint das Bereitschaftssymbol (>) in der Anzeige.

## AUSSCHALTEN DES COMPUTERS

Um den PC 1350 auszuschalten, bringen Sie den Schiebeschalter in die Position OFF.

Wenn Sie das Gerät auf "OFF" schalten, löschen Sie das Zeilendisplay. Die Programme und Daten bleiben jedoch erhalten und stehen zur Verfügung, sobald das Gerät wieder eingeschaltet wird.

Wenn der CLOAD-Befehl ausgeführt wird, unterbrechen Sie die Ausführung, indem Sie die **BRK**-Taste drücken und den Schiebeschalter in die Position OFF bewegen.

## AUTOMATISCHE ABSCHALTUNG

Um die Batterien vor unnützer Beanspruchung zu schützen, schaltet sich der PC 1350 automatisch ab, wenn 11 Minuten lang keine Taste betätigt wurde. (Achtung: Die automatische Abschaltung wird ausgesetzt, solange der PC 1350 ein Programm abarbeitet!)

Alle Daten und Programm bleiben bei automatischer Abschaltung gespeichert. Wenn Sie den Computer nach automatischer Abschaltung wieder starten wollen, tun Sie dies mit der **BRK**-Taste.

## EINIGE HILFREICHE HINWEISE

Bis Sie sich an Ihr neues Gerät gewöhnt haben, werden Sie sicherlich Fehler bei der Dateneingabe machen. Wir werden später einige einfache Korrekturmöglichkeiten kennenlernen. Für's erste: Wenn Sie eine Fehlermeldung erhalten, drücken Sie die rote CLEAR-Taste ( **CLS** ) und wiederholen die Eingabe. Wenn der Computer "sich aufhängt" - d. h. überhaupt nicht mehr reagiert - betätigen Sie den ALL-RESET-Schalter (siehe Kapitel 2).

Das Bereitschaftssymbol ( > ) sagt Ihnen, daß der PC 1350 eine Eingabe erwartet. Wenn Sie Daten eingeben, verschwindet das Bereitschaftssymbol und der Cursor bewegt sich nach rechts und zeigt dabei die nächstverfügbare Stelle im Zeilendisplay an.

Mit den Pfeilen nach rechts ( **▶** ) und links ( **◀** ) bewegen Sie den Cursor innerhalb einer Zeile.

Wenn Sie **ENTER** drücken, informieren Sie den PC 1350 darüber, daß Sie die Dateneingabe beendet haben und Sie fordern den Computer auf, die gewünschten Operationen durchzuführen. SIE MÜSSEN **ENTER** AM ENDE JEDER EINGABEZEILE BETÄTIGEN; ANDERNFALLS WERDEN IHRE RECHNUNGEN VOM COMPUTER NICHT AUSGEFÜHRT WERDEN!

Wenn Sie numerische Berechnungen durchführen, erscheint die Eingabe links im Display, das Ergebnis steht rechts im Display.

Benutzen Sie die SHIFT -Funktion in Verbindung mit einer anderen Taste (um z. B. die Quadratwurzel zu ziehen), drücken Sie **SHIFT**, lassen Sie **SHIFT** wieder los und drücken dann die andere Taste. Der SHIFT -Modus bezieht sich lediglich auf die nächste gedrückte Taste.

Benutzen Sie keine \$-Zeichen oder Kommata, wenn Sie mit dem PC 1350 ohne Programmunterstützung rechnen. Diese Zeichen haben eine besondere Bedeutung in der Programmiersprache BASIC.

In diesem Handbuch benutzen wir die Ø, um die Null zu kennzeichnen, so daß zwischen der Ziffer (0) und dem Buchstaben (O) unterschieden werden kann.

Um Ihnen die korrekte Dateneingabe am Anfang zu erleichtern, werden wir in den Beispielrechnungen jede nötige Tastenbedienung zeigen. Wenn die **SHIFT** -Funktion angesprochen wird, werden wir das gewünschte Zeichen im folgenden Tastensymbol abbilden. So entsteht beispielsweise durch Betätigen von **SHIFT** und **Q** das " ! "-Zeichen. Eine solche Tastenbedienung wird mit **SHIFT** **!** dargestellt.

Denken Sie daran, nach jeder Rechenoperation die **CLS** -Taste zu betätigen. (Es sei denn, Sie möchten Kettenrechnungen durchführen.) CLEAR löscht die Anzeige und beseitigt eventuelle Blockaden durch Fehler. Speicherinhalte werden nicht gelöscht.

## EINFACHE RECHENOPERATIONEN

Der PC 1350 rechnet auf 1 Stelle genau. Wenn Sie es nicht bereits getan haben, schalten Sie jetzt Ihren Computer ein und bringen Sie ihn den RUN-Modus. Nun versuchen Sie, die folgenden einfachen Rechenbeispiele nachzuvollziehen.

## EINGABE

5 0 + 5 0 ENTER

1 0 0 - 5 0 ENTER

6 0 \* 1 0 ENTER

3 0 0 / 5 ENTER

1 0 SHIFT ^ ENTER

2 \* SHIFT π ENTER

SHIFT √ 6 4 ENTER

## ANZEIGE

50 + 50	100.
---------	------

50 + 50	100.
100 - 50	50.

100 - 50	50.
60 * 10	600.

Das Display des PC 1350 besteht aus 4 Zeilen (je 24 Zeichen pro Zeile). Alle Daten und Rechenergebnisse werden jeweils von der obersten Zeile an ausgegeben. Wenn die Daten die Kapazität von 4 Zeilen überschreiten, wird der Anzeigehalt jeweils um 1 Zeile nach oben gerollt (die erste angezeigte Zeile rollt nach oben und verschwindet dann aus dem Display).

50 * 10	600.
300 / 5	60.

300 / 5	60
10 ^ 2	100.



10 ^ 2	100.
2 * π	6.283185307


2 * π	6.283185307
√ 64	8.


4 E 3 ENTER

$\sqrt{64}$	8.
4 E 3	4000.

## WIEDERABRUF VON EINGABEN

Auch nachdem der PC 1350 die Ergebnisse Ihrer Rechenoperationen angezeigt hat, können Sie Ihre vorherige Eingabe editieren lassen. Dazu benutzen Sie die Rechts-  und Links-  Pfeile.

Der Links-  -Pfeil wird benutzt, um den Cursor an die Stelle hinter dem letzten Zeichen zu positionieren.

Der Rechts-  -Pfeil wird benutzt, um den Cursor auf das erste Zeichen zu setzen.

Sie erinnern sich: der Rechts- und Links-Pfeil wird auch benutzt, um den Cursor innerhalb der Zeile zu bewegen. Die Pfeile sind sehr hilfreich beim Kontrollieren oder Verändern von Eingaben, ohne dabei den gesamten Ausdruck noch einmal eingeben zu müssen.

Sie werden mit dem Gebrauch der Rechts- und Links-Pfeile im Verlauf der folgenden Beispiele schnell vertraut werden. Schlüpfen Sie nun in die Rolle eines Managers und führen unsere Beispielrechnungen selbst durch, während wir sie erklären.


Als Personalchef in einem großen Marketing-Unternehmen sind Sie verantwortlich für die Planung der jährlichen Verkaufstagung. Sie erwarten zu diesem dreitägigen Treffen 300 Personen. Für einen Teil dieser Zeit wird man sich in kleine Arbeitsgruppen aufteilen. Sie sind der Ansicht, daß Gruppen zu je 6 Personen eine gute Größe wären. Wieviele Gruppen müssen Sie einplanen?

## EINGABE

## ANZEIGE

3 0 0 / 6 ENTER

300 / 6	50.
---------	-----

Dann fällt Ihnen ein, daß Gruppen mit einer ungeraden Teilnehmerzahl möglicherweise effektiver arbeiten. Rufen Sie Ihre letzte Eingabe mit Hilfe des  -Pfeils wieder auf.

EINGABEANZEIGE

300 / 6\_

Um die neue Anzahl der Gruppen zu berechnen, müssen Sie die 6 durch eine ungerade Zahl ersetzen. Fünf Gruppenmitglieder scheinen Ihnen sinnvoller als sieben. Da Sie daran gedacht haben, den -Pfeil zu benutzen, steht der Cursor nun am Ende der Anzeige. Benutzen Sie noch einmal den -Pfeil, um den Cursor eine Stelle nach links zu bewegen.

EINGABEANZEIGE

300 / 6

Beachten Sie, daß nach dieser Bewegung der Cursor sich in ein blinkendes Viereck verwandelt. Immer, wenn Sie den Cursor auf ein schon bestehendes Zeichen setzen, wird er in dieser Form dargestellt.

Geben Sie nun eine 5 anstelle der 6 ein. Achtung beim Ersetzen von Zeichen: Wenn Sie ein neues Zeichen über ein bereits bestehendes setzen, ist das ursprüngliche endgültig gelöscht, Sie können keinen Ausdruck wieder abrufen, der überschrieben wurde.

EINGABE

5 ENTER

ANZEIGE

300 / 5

60.

Sechzig scheint Ihnen eine vernünftige Anzahl von Gruppen zu sein, so beschließen Sie, daß jede Kleingruppe aus 5 Teilnehmern bestehen wird.

Erinnern Sie sich, daß es sinnvoll ist, die letzte Eingabe zu kontrollieren, besonders, wenn Ihre Ergebnisse Ihnen zweifelhaft erscheinen. Nehmen Sie z. B. an, Sie hätten folgende Rechnung durchgeführt:

EINGABE

CLS 300 / 5 ENTER

ANZEIGE

30 / 5



6.

Selbst ein müder, überarbeiteter Manager, wie Sie es sind, bemerkt, daß 6 kein glaubwürdiges Ergebnis ist, wenn man mit Hunderten von Personen operiert. Rufen Sie Ihre Eingabe mit Hilfe des -Pfeiles noch einmal ins Zeilendisplay.

EINGABEANZEIGE

30 / 5



Weil Sie den -Pfeil benutzt haben, steht der blinkende Cursor jetzt auf dem ersten Zeichen der Anzeige. Um die Eingabe zu korrigieren, möchten Sie jetzt eine zusätzliche Null einfügen. Mit Hilfe des -Pfeils bewegen Sie den Cursor, bis er über der Null steht. Wenn Sie nun mit der **INS**-Taste eine Einfügung machen wollen, setzen Sie den blinkenden Cursor auf denjenigen Buchstaben, vor dem das neue Zeichen eingefügt werden soll.

EINGABEANZEIGE

30 / 5

Benutzen Sie die **INS**-Taste, um Platz für das benötigte Zeichen zu schaffen.

EINGABE**INS**ANZEIGE

30 / 5

Das Betätigen der **INS**-Taste bewegt alle Zeichen um einen Schritt nach rechts und fügt einen von Klammern eingeschlossenen Leerraum ein. Der blinkende Cursor steht nun auf diesem Leerraum und zeigt die Stelle der nächsten Eingabe an. Geben Sie die fehlende Null ein. Nachdem die Eingabe korrigiert ist, rufen Sie erneut Ihr Ergebnis ab.

EINGABE **ENTER**ANZEIGE

300 / 5


60.

Stellen Sie sich auf der anderen Seite vor, Sie hätten die folgende Eingabe gemacht:

EINGABE**CLS** **3** **0** **0** **0** **/** **5** **ENTER**ANZEIGE


3000 / 5

600.

Das Ergebnis scheint viel zu groß. Wenn Sie nur 300 Personen zu Ihrem Treffen erwarten, wie könnte es dann 600 Arbeitsgruppen geben? Rufen Sie Ihre Eingabe mit Hilfe des -Pfeils ab.

EINGABEANZEIGE

3000 / 5

Der blinkende Cursor steht nun über dem ersten Zeichen der Anzeige. Um die Eingabe zu korrigieren, müssen Sie eine der Nullen löschen. Mit Hilfe des -Pfeils bewegen Sie den Cursor auf die erste (oder eine andere) Null. Wenn Sie ein Zeichen löschen wollen, setzen Sie den Cursor auf das zu löschende Zeichen.

EINGABEANZEIGE

3000 / 5

Nun betätigen Sie die **DEL** -Taste, um eine der Nullen zu beseitigen.

EINGABE**DEL**ANZEIGE

300 / 5

Durch Betätigen der **DEL** -Taste rutschen alle Zeichen um einen Schritt nach links. Das Zeichen, auf dem der Cursor steht, wird gelöscht, ebenso der Freiraum, den dieses Zeichen beanspruchte. Der blinkende Cursor bleibt an derselben Stelle stehen und zeigt die nächste Stelle für eine Eingabe ein. Da Sie keine weiteren Veränderungen vorzunehmen haben, beenden Sie die Rechnung.

EINGABE**ENTER**ANZEIGE

300 / 5

60.

(Achtung: Steht der Cursor auf einem Zeichen und Sie betätigen die **SPC** -Taste, wird das betreffende Zeichen gelöscht und durch einen Leerschritt ersetzt. Mit **DEL** löschen Sie das Zeichen und den von ihm eingenommenen Leerraum.)

FEHLER

Der Abruf der vorherigen Eingabe ist notwendig, wenn Sie die gefürchtete ERROR-Meldung erhalten. Wir wollen uns vorstellen, daß Sie unabsichtlich folgendes in den PC 1350 eingegeben haben:

EINGABE**CLS** **3** **0** **0** **/** **/** **5** **ENTER**ANZEIGE300 / / 5  
ERROR 1

Natürlich sind Sie überrascht, wenn diese Meldung erscheint! ERROR 1 ist einfach die Art Ihres Computers, Ihnen mitzuteilen: "Ich weiß nicht, was ich hier tun soll."

Wenn Sie in diesem Fall den -Pfeil betätigen, erscheint der blinkende Cursor an der Stelle, wo der Fehler liegt.

EINGABEANZEIGE

300 / 7 5

Um diesen Fehler zu korrigieren, benutzen Sie die **DEL** -Taste.

EINGABE**DEL** **ENTER**ANZEIGE

300 / 5

60.

Wenn Sie beim Abruf Ihrer Eingabe aufgrund der Meldung **ERROR 1** feststellen, daß Sie ein Zeichen vergessen haben, benutzen Sie die **INS** -Sequenz, um es zu korrigieren.

Wenn Sie den PC 1350 wie einen normalen Taschenrechner benutzen, werden Sie in der Hauptsache mit dem **ERROR 1** (Syntaxfehler) konfrontiert werden. Eine komplette Aufstellung der Fehlermeldungen und ihrer Bedeutungen finden Sie in Anhang A.

VERKETTUNG VON RECHENOPERATIONEN

Ihr PC 1350 erlaubt, daß Sie das Ergebnis einer Rechnung in der folgenden Operation weiterverwenden.

Zurück zu unserem Beispiel: Ein Teil Ihrer Verantwortung bei der Planung dieser Konferenz besteht darin, einen detaillierten Budget-Vorschlag auszuarbeiten. Sie wissen, daß Ihnen für jeden Teilnehmer DM 150,- zur Verfügung stehen. Berechnen Sie Ihr Gesamtbudget.

EINGABE**CLS** **3** **0** **0** **\*** **1** **5** **0** **ENTER**ANZEIGE

300 \* 150

45000.

Von dieser Summe beabsichtigen Sie, 15 % für die Abschlußfeier aufzuwenden. Wenn Sie verkettete Rechenoperationen durchführen, ist es nicht nötig, Ihr erstes Ergebnis noch einmal einzugeben, wenn Sie die **CLS**-Taste NICHT benutzen. Wie hoch ist Ihr Budget für die Feier?

EINGABE**\*** **.** **1** **5** **ENTER**ANZEIGE

300 \* 150

45000.

45000. \* .15

6750.

Beachten Sie, daß der Computer das Ergebnis Ihrer ersten Berechnung automatisch links im Zeilendisplay anzeigt und in die neue Rechnung miteinbezieht, wenn Sie die Befehle für die zweite Berechnung eingeben. In verketteten Rechenoperationen muß die Eingabe mit einem Operationssymbol beginnen. Beenden Sie Ihre Eingabe wie immer mit **ENTER** -Taste.

Achtung: Die  $\%$ -Taste kann nicht in Rechenoperationen benutzt werden. Sie sollte ausschließlich als ein Zeichen benutzt werden.

Beispiel:

EINGABE

CLS 4 5 0 0 0 \* 1  
5 SHIFT % ENTER

ANZEIGE

45000 \* 15 %  
ERROR 1

Fahren Sie fort, Ihr Budget zu berechnen. Das Hotel verlangt für die Verpflegung DM 4000,--.

EINGABE

- 4 0 0 0 ENTER

ANZEIGE

45000 \* .15  
6750. - 4000 6750.  
2750.

Die Dekoration wird DM 1225,-- kosten.

EINGABE

- 1 2 2 5 ENTER

ANZEIGE

6750. - 4000  
2750. - 1225 2750.  
1525.

Schließlich müssen Sie DM 2.200,-- für den Sprecher und für das Rahmenprogramm einkalkulieren.

EINGABE

- 2 2 0 0 ENTER

ANZEIGE

2750. - 1225  
1525. - 2200 1525.  
-675.

Nun, Sie werden entweder Ihre Pläne oder Ihre Kostenverteilung ändern müssen!

### NEGATIVE ZAHLEN

Da die Abschlußfeier etwas ganz Besonderes sein soll, beschließen Sie, Ihre Pläne nicht zu ändern und das zusätzliche Geld zu investieren. Trotzdem fragen Sie sich, wieviel Prozent des Gesamtbudgets wohl von diesem Posten in Anspruch genommen werden. Zuerst ändern Sie das Vorzeichen Ihres letzten Ergebnisses.

EINGABE

ANZEIGE

\* - 1 ENTER

1525. - 2200	
	-675.
-675. * - 1	
	675.

Nun addieren Sie dieses Ergebnis zu Ihrem ursprünglichen Budget für die Feier hinzu.

EINGABE

ANZEIGE

+ 6 7 5 0 ENTER

-675. * - 1	
	675.
675. + 6750	
	7425.

Teilen Sie Ihr neues Ergebnis durch 45000 und Sie erfahren, wieviel Prozent des Gesamtbudgets diese neue Zahl ausmacht.

EINGABE

ANZEIGE

/ 4 5 0 0 0 ENTER

675. + 6750	
	7425.
7425. / 45000	
	0.165

Nun gut, Sie beschließen, 16,5 % für die Abschlußfeier aufzuwenden.

**KOMPLEXERE RECHENOPERATIONEN UND KLAMMERUNG**

Bei der Durchführung der obigen Rechenbeispiele hätten Sie einige dieser Operationen in einem Schritt zusammenfassen können. So hätten Sie etwa die beiden folgenden Berechnungen in eine Zeile schreiben können:

$$675 + 6759 / 450000$$

Solche komplexeren Rechnungen müssen jedoch sehr sorgfältig eingegeben werden.

675 + 6750 / 450000 könnte verstanden werden als

$$\frac{675 + 6750}{45000} \quad \text{oder} \quad 675 + \frac{6750}{45000}$$

Bei der Durchführung komplexer mathematischer Operationen folgt der PC 1350 spezifischen Regeln der impliziten Klammerung und der Rangfolge von mathematischen Operatoren (siehe Anhang D). Um sicherzustellen, daß Ihre Eingaben in der von Ihnen gewünschten Reihenfolge verarbeitet werden, sollten Sie Klammern verwenden.

$$(675 + 6750) / 45000 \quad \text{oder} \quad 675 + (6750 / 45000)$$

Wie unterschiedlich sich die Platzierung von Klammern auf das Ergebnis auswirken kann, sehen Sie in den beiden folgenden Beispielen:

EINGABE

( 6 7 5 + ) 6  
 7 5 0 ) 7 4  
 5 0 0 0 ENTER

CLS 6 7 5 + ( 6  
 7 5 0 7 4 5 0 0 0  
 0 ) ENTER

ANZEIGE

(675 + 6750) / 45000

0.165

675 + (6750 / 45000)

675.15

### DER GEBRAUCH VON VARIABLEN IN RECHENOPERATIONEN

Der PC 1350 kann bis zu 26 einstellige numerische Variablen unter den alphabetischen Zeichen A - Z speichern. Wenn Ihnen der Begriff der Variablen nicht vertraut ist: In Kapitel 4 finden Sie ausführliche Erklärungen. Die Variablenzuweisung geschieht über einen Zuweisungsbefehl:

$$A = 5$$

$$B = -2$$

Sie können auch den Wert einer Variablen (rechts) einer anderen (links) zuweisen.

$$C = A + 3$$

$$D = E$$

Eine Variable kann in jeder Rechenoperation anstelle einer Zahl benutzt werden.

Nachdem Sie nun Ihre Abschlußfeier geplant haben, fehlen Ihnen noch die übrigen Berechnungen für Ihre Verkaufstagung. Sie möchten den Rest Ihres Budgets ebenfalls prozentual aufteilen. Zuerst müssen Sie herausfinden, wieviel Geld Sie noch zur Verfügung haben. Weisen Sie der Variablen (den verbleibenden Rest) zu.

EINGABE

R = 4 5 0 0 0 - 7 4 2 5  
 7 4 2 5 ENTER

ANZEIGE

R = 45000 - 7425

37575.

Wenn Sie nun **ENTER** drücken, führt der PC 1350 die Berechnung durch und zeigt den neuen Wert von R an. Sie können den Wert jeder Variablen zur Anzeige bringen, indem Sie den Buchstaben eingeben, unter dem er gespeichert ist.

EINGABE

R ENTER

ANZEIGE

R = 45000 - 7425	37575.
R	37575.

Nun können Sie Rechenoperationen mit Ihren Variablen durchführen. Der Wert von R wird sich nicht verändern, bis Sie ihm einen neuen Wert zuweisen.

Sie beabsichtigen, 60 % des verbleibenden Geldes auf die Unterbringung zu verwenden.

EINGABE

CLS R \* . 6 0  
ENTER

ANZEIGE

R * .60	22545.
---------	--------

Weiterhin wollen Sie 25 % des verbleibenden Budgets für die Durchführung von Management-Trainings-Seminaren ausgeben.

EINGABE

CLS R \* . 2 5 ENTER

ANZEIGE

R * .25	9393.75
---------	---------

Variable behalten ihren zugewiesenen Wert auch, wenn das Gerät ausgeschaltet wird oder sich automatisch abschaltet.

Sie gehen nur verloren, wenn

- man ihnen einen neuen Wert zuweist,
- man CLEAR ENTER (nicht die CLS -Taste) eingibt,
- man das Gerät mit dem ALL-RESET-Schalter initialisiert,
- die Batterien gewechselt werden.

Es gibt gewisse Grenzen der Variablenzuweisung sowie gewisse Programmabläufe, die Variablen verändern können. In Kapitel 4 finden Sie detaillierte Informationen zur Variablenzuweisung, in Kapitel 5 wird der Gebrauch von Variablen bei der Programmierung erörtert.

**MEHRERE RECHENSCHRITTE**

Mit dem PC 1350 ist es nicht nur möglich, verschiedene Rechenschritte zusammenzufassen, man kann auch einzelne Rechnungen nacheinander durchführen, ohne zwischendurch ENTER betätigen zu müssen. Die einzelnen Rechenoperationen werden durch Kommata getrennt. In der Anzeige erscheint nur das Ergebnis der letzten Rechenoperation. (Beachten Sie bitte, daß die maximale Zeilenlänge, die Ihr Computer annimmt, 80 Zeichen einschließlich ENTER beträgt.)

Sie fragen sich, wieviel Geld für die Unterbringung zur Verfügung stünde, wenn Sie, wie ursprünglich geplant, nur 15 % für die Abschlussfeier aufwenden würden.

EINGABE

R, =, ., 8, 5, \*, 4, 5  
 0, 0, 0, R, \*, ., 0, 0

ANZEIGE

R = .85 \* 45000, R \* .60

Obwohl der Computer alle eingegebenen Rechenoperationen durchführt, zeigt er nur das letzte Ergebnis an.

EINGABE

ENTER

ANZEIGE

R = .85 \* 45000, R \* .60

22950.

Um herauszufinden, welchen Wert die Variable R in dieser Berechnung hatte, geben Sie R ein:

EINGABE

CLS R ENTER

ANZEIGE

R

38250.

WISSENSCHAFTLICHE NOTATION

Wer mit sehr großen und sehr kleinen Zahlenwerten zu arbeiten hat, benutzt oft ein spezielles Format, die sogenannte exponentielle oder wissenschaftliche Notation. In der wissenschaftlichen Notation wird eine Zahl in zwei Teile aufgespalten.

Der erste Teil besteht aus einer Dezimalzahl zwischen 1 und 10. Der zweite Teil gibt die Größe der Zahl in Zehnerpotenzen an.

Wie Sie wissen, ist in einer gewöhnlichen Dezimalzahl die erste Stelle links vom Komma die Stelle der Einer, an zweiter Stelle folgen die Zehner, an dritter die Hunderter, dann die Tausender usw. Dies sind einfach ansteigende Potenzen von 10.

$$10^0 = 1, \quad 10^1 = 10, \quad 10^2 = 100, \quad 10^3 = 1000 \text{ usw.}$$

Die wissenschaftliche Notation bricht also eine Dezimalzahl in zwei Teile auseinander. Ein Teil zeigt, um welche Ziffernfolge es sich handelt, der zweite, wieviele Stellen eine Zahl links oder rechts vom Komma aufweist. Z. B.:





## EINGABE

3 + 4 ENTER

\* 5 ENTER

## ANZEIGE

3 + 4	7.
3 + 4	7.
7. * 5_	35.

Drücken Sie **CLS**, dann die **↑**- oder **↓**-Taste. Wenn Sie diese Tasten unmittelbar nach Abschluß der obigen Berechnung bedient haben, sollten Sie jetzt "35" in der Anzeige sehen. Die ausgegebene Zahl ist das Ergebnis Ihrer letzten Berechnung.

Der PC 1350 "merkt sich" das letzte ermittelte Ergebnis, und mit Hilfe der **↑**- oder **↓**-Taste kann es wieder abgerufen werden.

Im Fall der oben beschriebenen Kettenrechnung konnten Sie das Ergebnis der vorherigen Rechenoperation ausschließlich als erstes Glied des folgenden Ausdrucks benutzen. Mit Hilfe der Werte-Schablone jedoch können Sie das Ergebnis der vorherigen Berechnung an jeder beliebigen Stelle in die anschließende Rechenoperation einfügen.

Beispiel:

Benutzen Sie das Ergebnis (6.25) der Rechnung  $50 / 8$  um zu berechnen:  
 $12 \times 5 / 6.25 + 24 * 3 / 6.25 =$

## EINGABE

5 0 / 8 ENTER

## ANZEIGE

50 / 8	6.25
--------	------

CLS 1 2 \* 5 / ↑

12 * 5 / 6.25_
----------------

+ 2 4 \* 3 / ↓

12 * 5 / 6.25 + 24 * 3 / 6.25
-------------------------------

ENTER

12 * 5 / 6.25 + 24 * 3 / 6.25	21.12
-------------------------------	-------

CLS ↓

21.12_
--------

Wie in diesem Beispiel gezeigt, kann das letzte Ergebnis jederzeit und an jeder beliebigen Stelle abgerufen werden, es wird jedoch ersetzt, sobald eine neue Berechnung ein neues Endergebnis liefert.

Das gespeicherte Ergebnis wird durch Betätigen der CLS -Taste nicht gelöscht.

\* Das letzte Ergebnis kann nicht abgerufen werden, wenn sich der Computer nicht im RUN-Modus befindet, wenn die Programmausführung vorübergehend unterbrochen ist oder wenn der Trace-Modus gewählt wurde.

### LÄNGE VON AUSDRÜCKEN

Die Länge von Ausdrücken, die Sie in Ihren Computer eingeben können, unterliegt gewissen Begrenzungen. Mit dem PC 1350 können bis zu 79 Tastenbedienungen für einen einzigen Rechenausdruck verwendet werden (ausgenommen die ENTER -Taste). Wenn Sie ein 80. Zeichen eingeben wollen, wird der Cursor (   ) an dieser Stelle zu blinken beginnen und damit anzeigen, daß die 80. Eingabe nicht gewertet wird.

### TABELLE FÜR DIE WISSENSCHAFTLICHEN FUNKTIONEN

Der PC 1350 ist mit den in dieser Tabelle angegebenen mathematischen Basisfunktionen ausgestattet. Beachten Sie bitte, daß die Darstellung in BASIC sich von der normalen mathematischen Schreibung unterscheiden kann.

FUNKTION	mathematische Schreibweise	BASIC Befehl	Bemerkungen
trigonometische Funktionen	sin cos tan	SIN COS TAN	
trigonometische Umkehrfunktionen	arcsin arccos arctan	ASN ACS ATN	
10er Logarithmus	log	LOG	Logarithmus zur Basis 10
natürlicher Logarithmus	ln	LN	Logarithmus zur Basis e
Exponentialfunktion	e	EXP	e=2.718281826

FUNKTION	mathematische Schreibweise	BASIC Befehl	Bemerkungen
Potenzierung	$\begin{matrix} b \\ a \end{matrix}$	$\wedge$	$a^b = a$ hoch $b$
Quadratwurzel	$\sqrt{\quad}$	$\sqrt{\quad}$ oder SQR	
Umwandlung Dezimal-Grad nach Winkel-Grad (Grad, Minute, Sekunde)		DMS	bei Winkel-Umwandlung darf nicht wie in DEG .5 die 0 wegfallen
Umwandlung Winkel-Grad (Grad, Minute, Sekunde) nach Dezimal-Grad		DEG	
ganze Zahlen		INT	INT(x) ergibt die größte ganze Zahl kleiner oder gleich x
Betrag	$ x $	ABS	ABS(x) ergibt den Betrag von x
Vorzeichen		SGN	SGN(x) = 1 f. $x > 0$ = -1 f. $x < 0$ = 0 f. $x = 0$
Pi	$\pi$	$\pi$ oder PI	$\pi = 3.141592654$
Hexadezimal $\longrightarrow$ dezimale Schreibweise		&	Wandelt eine hexadezimale Zahl x auf der Basis 10 (&x) um.

Winkel einheit	Befehl		Erklärung
Alt-Grad	DEGREE		Rechter Winkel = 90 Grad
Bogenmaß	RADIAN		Rechter Winkel = $\pi/2$ rad.
Neugrad	GRAD		Rechter Winkel = 100

Diese Befehle werden benutzt, um Winkelmaße in Programmen zu spezifizieren. Benutzen Sie diese Befehle zur Übung in den folgenden Rechenbeispielen:

- $\sin 30^\circ =$   
 DEGREE   
 SIN 30  0.5
- $\tan \frac{\pi}{4} =$   
 RADIAN   
 TAN (PI/4)  1.
- $\cos^{-1}(-0.5) =$   
 DEGREE   
 ACS -0.5  120.
- $\log 5 + \ln 5 =$   
 LOG 5 + LN 5  2.308407917
- $e^{2+3} =$   
 EXP (2 + 3)  148.4131591
- $\sqrt[4]{3^4 + 6^4} =$   
 $\sqrt{(4^3 + 6^4)}$   36.87817783

7. Wandle  $30$  Grad  $30'$  in DMS-Darstellung um in Dezimal-Darstellung (DEG).

DEG 30.30  | 30.5 |  
(30,5 Grad)

8. Wandle  $30.755$  in Dezimaldarstellung (DEG) um in DMS-Darstellung.

DMS 30.755  | 30.4518 |  
(30 ,45min,18sec)

9. Wandle CF8 in eine Dezimalzahl um.

&CF8  | 3320. |

### RANGFOLGE VON OPERATIONEN

Sie können Ausdrücke in derselben Reihenfolge eingeben, in der sie geschrieben sind. Die interne Rangfolge bei der Berechnung und die Behandlung von Zwischenergebnissen wird vom Computer kontrolliert.

Die interne Rangfolge beim Rechnen ohne Programmunterstützung ist die folgende:

- 1) Abruf von Variablen oder Pi
- 2) Funktionen (sin, cos etc.)
- 3) Exponentiation (^)
- 4) Vorzeichen (+, -)
- 5) Multiplikation oder Division (\*, /)
- 6) Addition oder Subtraktion (+, -)
- 7) Größenvergleich (>, >=, <, <=, <>)
- 8) Logisches AND, OR

#### Anmerkungen:

- Werden in einem Ausdruck Klammern verwendet, hat die innerhalb der Klammern angegebene Operation höchste Priorität.
- Zusammengesetzte Funktionen werden von rechts nach links abgearbeitet (sin cos <sup>-1</sup> 0.6).
- Mehrstufige Exponentiation wird von rechts nach links abgearbeitet  

$$(3^4)^2 \text{ oder } 3^{4^2}.$$
- Von den obigen Punkten 3) und 4) hat die letzte Eingabe die höhere Priorität.  

$$\text{z. B. } -2^4 \rightarrow -(2^4)$$

$$3^{-2} \rightarrow 3^{-2}$$

DIE AUSGABE VON RECHENERGEBNISSEN

Die Ergebnisse Ihrer Berechnungen werden normalerweise auf dem Display angezeigt. Sie können jedoch wahlweise - sofern die optionale Einheit CE-126P angeschlossen und eingeschaltet ist - durch aufeinanderfolgendes Drücken der Tasten **SHIFT** und **P↔PN** eine Ausgabe auf dem Drucker erreichen (PRINT-Modus).

Wenn der Ausdruck nicht gewünscht wird, schalten Sie wahlweise die Option CE-126P aus oder drücken **SHIFT** und **P↔NP** erneut (NON-PRINT Modus).

RECHENFEHLER

Folgende Fehlertypen treten bei gewöhnlichen Rechnern, Taschen- und Personalcomputern auf:

- (1) Fehler, die mit der Anzahl der Kommastellen zusammenhängen

Normalerweise ist die maximale Anzahl der Dezimalstellen, mit denen ein Computer rechnen kann, festgelegt. Beispiel: 4 ./ 3 ergibt 1.33333333... Bei einem Computer mit einem Maximum von 8 Stellen werden diese 8 Ziffern als signifikant bezeichnet, darüberhinausgehende Ziffern werden entweder ignoriert oder gerundet.

BEISPIEL: Computer mit 10 signifikanten Stellen

10 signifikante Ziffern

4 / 3 **ENTER**

1.333333333333...

↑  
— ignoriert, gerundet

Aus diesem Grunde weicht das errechnete Ergebnis von dem tatsächlichen Ergebnis durch den ignorierten oder gerundeten Betrag ab. (Dieser Unterschied macht dann die Abweichung aus.)

Der Rechner ermittelt ein 12stelliges Resultat. Das Ergebnis wird gerundet und so verarbeitet, daß die Fehlerquote so weit als möglich minimiert wird und das angezeigte Ergebnis dem tatsächlichen weitgehend nahekommt.

BEISPIEL: 4 ./ 3 \* 3

(1) 4 / 3 \* 3 **ENTER**

4.

wird sukzessive  
verarbeitet

(2) 4 / 3 **ENTER**

1.3333333333

wird unabhängig  
verarbeitet

\* 3 **ENTER**

3.999999999

- \* Während der Rechner die Operation sukzessive verarbeitet, erhält er als Ergebnis eine Ziffer mit 12 Kommastellen, die er als Resultat akzeptiert und anschließend rundet.

Wenn der Rechner unabhängig arbeitet, wird der angezeigte Wert (10 Kommastellen) für die Berechnung benutzt.

- (2) Fehler, die bei der sukzessiven Verarbeitung auftreten

Obwohl der Rechner während der Rechenoperationen die Abweichung des angezeigten Wertes so weit als möglich zu minimieren versucht, summieren sich jedoch im Laufe einer längeren Berechnung diese Abweichungen und erscheinen als Fehler auf dem Display.

BEISPIEL:  $60^6 =$

$60^6$    $\longrightarrow 4.665599999E10$

Obwohl  $60^6$   $4.6656 \cdot 10^{10}$  ergibt, geht der Rechner von einem Wert ( $y$ ) aus, der dem folgenden Ausdruck entspricht:

$$y = 10^{x \cdot \log y}$$

Mit anderen Worten:  $60^6$  wird als  $10^{6 \cdot \log 60}$  berechnet.

Obwohl die Funktion selbst einen Näherungsalgorithmus verwendet, ist der Fehler im Computer minimal größer als der bei regulären Berechnungen. Daher summieren sich diese Abweichungen im Verlauf einer längeren Rechenoperation und können auf dem Display erscheinen.



## KAPITEL 4

## BEGRIFFE UND AUSDRÜCKE DES BASIC

In diesem Kapitel wollen wir einige Begriffe und Ausdrücke der Programmiersprache BASIC kennenlernen.

## STRING-KONSTANTEN

Der PC 1350 ist in der Lage, außer Zahlen auch Buchstaben und spezielle Symbole in vielfacher Weise zu verarbeiten. Diese Buchstaben, Zahlen und speziellen Symbole werden Zeichen genannt. Folgende Zeichen stehen dem PC 1350 zur Verfügung:

```

1 2 3 4 5 6 7 8 9 0
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
! " # $ % & ( ) * + , - . / : ; < = > ? @  $\sqrt{\pi}$  ^
```

Im BASIC wird eine Gruppe von Zeichen als String bezeichnet. Damit der PC 1350 den Unterschied zwischen einem String und anderen Programmteilen, wie z. B. Befehlen oder Variablen-Bezeichnungen erkennen kann, muß man die Zeichen, die zu einem String gehören, in Anführungszeichen (") einschließen.

Es folgen einige Beispiele für String-Konstanten:

"HALLO"

"TSCHUESS"

"SHARP PC 1350"

Die folgenden Beispiele werden als String-Konstanten nicht akzeptiert:

"COMPUTER                      Anführungszeichen am Ende fehlen.

"IST" NICHT"                   Anführungszeichen dürfen nicht innerhalb  
eines Strings benutzt werden.

## HEXADEZIMALZAHLEN

Das Dezimalsystem ist nur eines von verschiedenen Zahlensystemen. Ein anderes, dessen Bedeutung im Zusammenhang mit Computern stark angewachsen ist, ist das Hexadezimalsystem. Das Hexadezimalsystem basiert auf der Zahl 16 statt auf der 10. Um hexadezimale Ziffern zu schreiben, benutzt man die Ziffern 0 - 9 sowie 6 weitere "Ziffern": A, B, C, D, E und F. Diese entsprechen den Zahlen 10, 11, 12, 13, 14, 15 und 16. Wenn Ihr PC 1350 eine Zahl als hexadezimal auffassen soll, setzen Sie ein UND-Zeichen (&) vor den betreffenden Ausdruck.

Hexadezimal		Dezimal
&A	=	10
&10	=	16
&100	=	256
&FFFF	=	65535

## VARIABLE

Computer sind aufgebaut aus einer Vielzahl von kleinsten Speichereinheiten, genannt Bytes. Jedes Byte kann man sich als einzelnes Zeichen vorstellen. Das Wort "Byte" erfordert beispielsweise vier Speicherbytes, weil es vier Buchstaben enthält. Um herauszufinden, wieviele Bytes zum Arbeiten zur Verfügung stehen, geben Sie einfach MEM **ENTER** ein. Die angezeigte Zahl gibt an, wieviele Bytes zum Programmieren frei sind.

Zahlen werden in codierter Form gespeichert. Aufgrund dieser Codierung ist der Computer in der Lage, auch lange Zahlen in nur 8 Bytes zu speichern. Die größtmögliche Zahl, die gespeichert werden kann, ist +9.999999999 E +99.

Die kleinste speicherbare Zahl ist +1.E-99. So erhalten Sie einen recht großen Zahlenbereich, innerhalb dessen Sie arbeiten können. Wenn jedoch das Ergebnis einer Rechnung diesen Rahmen übersteigt, wird Ihr Computer Ihnen dies mitteilen, indem er eine Fehlermeldung ausgibt (ERROR 2). Eine Erklärung dieser Fehlermeldung finden Sie in Anhang A. Geben Sie nun zur Probe ein.

9E99 \* 9 **ENTER**

Damit Ihr Computer wieder normal weiterarbeiten kann, brauchen Sie lediglich die **CLS** -Taste zu drücken.

Aber wie speichert man denn nun so viel Information? Das ist ganz einfach! Der Computer zieht es vor, verschiedenen Datengruppen Namen zu geben.

Wir wollen die Zahl 556 abspeichern. Sie können diese Zahl nennen, wie Sie möchten, aber für dieses Beispiel wollen wir ihr den Namen R geben. Die Anweisung LET kann benutzt werden, um den Computer zu veranlassen, einer Variablen einen Wert zuzuweisen - allerdings nur in einem Programmbefehl. Der LET-Befehl ist aber nicht unbedingt erforderlich, darum werden wir ihn nicht sehr oft benutzen. Geben Sie nun R = 556 ein und drücken Sie **ENTER**. Der Computer hat nun den Wert 556 mit dem Buchstaben R in Verbindung gebracht. Diese Buchstaben, die man benutzt, um Information zu speichern, nennt man Variable. Um den Inhalt der Variablen R abzufragen, drücken Sie die **CLS** -Taste, dann die Tasten **R** und **ENTER**. Der Computer antwortet, indem er den Wert 556 rechts im Display anzeigt. Diese Möglichkeit kann sehr nützlich sein, wenn man Programme oder Formeln eingibt.

Wir wollen als nächstes die Variable R in einer einfachen Formel verwenden. In dieser Formel steht R für den Radius eines Kreises, dessen Fläche wir berechnen wollen. Die Formel für die Kreisfläche


lautet:  $A = \pi * R^2$ . Geben Sie ein:        . Das Ergebnis ist 971179.3866. Dieses Verfahren des Gebrauchs von Variablen wird eingehender erklärt, wenn wir uns mit dem Programmieren beschäftigen.

Bislang haben wir uns nur mit numerischen Variablen befaßt. Wie speichert man nun alphabetische Zeichen? Grundsätzlich ist das Prinzip das gleiche, aber damit der Computer den Unterschied zwischen den beiden Variablen-Typen erkennen kann, muß nun ein \$ zum Namen der Variablen gesetzt werden. Wir wollen z. B. das Wort BYTE unter der Variablen B\$ speichern. Sehen Sie das \$-Zeichen hinter dem B? Dies sagt dem Computer, daß der Inhalt der Variablen B alphanumerisch bzw. ein Daten-String ist.

Damit dies klarer wird, geben Sie ein:        . Das Wort BYTE ist nun unter der Variablen B\$ gespeichert. Um sich zu vergewissern, drücken Sie die  -Taste und geben dann ein:    . In der Anzeige steht BYTE, diesmal aber auf der linken Seite der Anzeige.

Beachten Sie: Der Inhalt von Zeichen-Strings oder Variablen wird von der linken Ecke der ersten Zeile an angezeigt.

Die Variablen, mit denen der SHARP PC 1350 arbeitet, sind folgendermaßen aufgliedert:

Variable		numerische Variable	vorgegebene numerische Variable (A bis Z) einfache numerische Variable (AB, C1 usw.) numerische Feld-Variablen
		String-Variablen	vorgegebene String-Variablen (A\$ bis Z\$), einfache String-Variablen (BB\$, C2\$ usw.), String-Feld-Variablen

VORGEGEBENE VARIABLE

Der erste Bereich, die vorgegebenen Variablen, werden vom Computer grundsätzlich benutzt, um Daten zu speichern. Man kann ihn sich als reservierten Bereich denken. Mit anderen Worten: egal, wieviel Speicherplatz Ihr Programm in Anspruch nimmt, Ihnen stehen immer mindestens 26 Variablen zur Speicherung von Daten offen. Diese Daten können entweder NUMERISCH oder in STRING (Buchstaben) -Form gewählt werden. Vorgegebene Speicherplätze haben eine Kapazität von 8 Byte und können immer nur einen Datentyp zur Zeit aufnehmen. Geben Sie zur Erklärung ein:

A = 123

A\$

Sie erhalten die Meldung:

ERROR 9

Das bedeutet, daß Sie numerische Daten in den Speicherbereich mit dem Namen A gelegt haben und daß Sie dann dem Computer befohlen haben, diese Information als String-Daten wieder abzurufen. Das aber verwirrt den Computer, so daß er eine Fehlermeldung ausgibt. Drücken Sie die **[CLS]**-Taste, um die Fehlermeldung zu beseitigen. Versuchen Sie nun das folgende Beispiel:

A\$ = "ABC" **[ENTER]**

A **[ENTER]**

Wieder ist der Computer verwirrt und meldet ERROR 9. In der folgenden Abbildung können Sie sehen, daß die Variable A im selben Speicherbereich liegt wie die Variable A\$, daß B sich mit B\$ deckt usw.

Abbildung:

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)

## EINFACHE VARIABLE

Einfache Variablenbezeichnungen werden durch zwei (oder mehr) alphanumerische Zeichen charakterisiert, wie z. B. AA oder B1. Anders als feste Variable haben die einfachen Variablen keinen fest im Speicher reservierten Bereich. Der Speicherbereich für einfache Variable wird automatisch im Programm oder im Datenbereich bereitgestellt, sobald eine einfache Variable erstmalig benutzt wird.

Da für einfach numerische und einfache String-Variable verschiedene Speicherbereiche vorgesehen sind, können Variable mit dem gleichen Namen (wie z. B. AB und AB\$) gleichzeitig benutzt werden.

Wenn man alphanumerische Bezeichnungen für Variablen benutzen will, muß man als erstes Zeichen grundsätzlich einen Buchstaben (Sie können stets nur Großbuchstaben benutzen!) wählen. Werden mehr als zwei Buchstaben zur Definition eines Variablen-Namens benutzt, sind nur die beiden ersten Zeichen von Bedeutung.

## Achtung:

- \* Die Namen von Funktionen oder BASIC-Befehlen dürfen auf dem PC 1350 nicht zur Bezeichnung von Variablen benutzt werden (z. B. PI, IF, TO, ON, SIN usw.).
- \* Unter einer einzelnen String-Variablen können bis zu 16 Buchstaben oder Zeichen gespeichert werden.

## FELD-VARIABLE

In einigen Fällen ist es sinnvoll, Zahlen in organisierten Gruppen zu verarbeiten, z. B. eine Tabelle der Fußball-Ergebnisse oder eine Steuertabelle. Im BASIC werden solche Gruppen FELDER genannt. Ein Feld kann eindimensional sein, wie z. B. eine Liste, es kann aber auch zweidimensional sein, wie z. B. eine Tabelle.

Um ein Feld zu definieren, benutzt man den DIM-Befehl (Kürzel für Dimension). Felder müssen vor Gebrauch immer definiert werden! (Dies war nicht der Fall bei den Einwert-Variablen, die wir bislang benutzt haben.)

Die Form für die DIMENSIONIERUNG numerischer Felder ist:

DIM variablenname (größe)

Dabei bedeutet:

- |               |   |
|---------------|---|
| variablenname | Eine Bezeichnung der Variablen gemäß den oben besprochenen Benennungsregeln für numerische Variable.  |
| größe         | ist die Anzahl der Speicherplätze, die sich im Bereich zwischen 0 und 255 bewegen muß. Beachten Sie, daß, wenn Sie eine Größenordnung eingeben, ein Speicherplatz mehr bereitgestellt wird. |

Beispiele für mögliche DIMensionierungsbefehle sind:

```
DIM X(5)
DIM AA(24)
DIM Q5(Ø)
```

Der erste Befehl schafft ein Feld X mit 6 Speicherplätzen. Der zweite Befehl baut ein Feld mit 25 Speicherplätzen auf, der dritte ein Feld Q5 mit einem Speicherplatz, was im Grunde unsinnig ist (zumindest für Zahlen), da man ebensogut eine einwertige numerische Variable definieren könnte.

Es ist wichtig zu wissen, daß eine Feld-Variable X und eine Variable X sich unterscheiden und auch vom SHARP PC 1350 unterschieden werden können. Das erste X bezeichnet eine Serie von numerischen Speicherplätzen, das zweite einen einzelnen und getrennten Speicherplatz.

Nachdem Sie nun wissen, wie man Felder aufbaut, mögen Sie sich fragen, wie man die einzelnen Speicherplätze anspricht. Da die gesamte Gruppe unter einem einzigen Namen abgelegt ist, sprechen wir einen einzelnen Speicherplatz ("Element" genannt) an, indem wir an den Namen der Gruppe eine Zahl in Klammern anschließen. Diese Zahl wird "Index" genannt. So müßte man z. B., um die Zahl 9 an fünfter Stelle in unserem (vorher definierten) Feld X unterzubringen, schreiben:

$$X(4) = 9$$

Wenn Sie die Zahl 4 verwirrt, bedenken Sie, daß die Numerierung der Elemente in einem Feld mit Null beginnt und dann bis zu der definierten Obergrenze fortläuft.

Der besondere Vorteil von Feldern liegt in der Möglichkeit, einen längeren Ausdruck oder eine Variable als Index zu benutzen.

Um ein String-Feld zu DIMensionieren, muß man etwas anders vorgehen als bei numerischen Feldern:

DIM stringvariablenname (größe) \* länge

Dabei bedeutet:

stringvariablenname Eine Bezeichnung der Variablen gemäß den oben besprochenen Benennungsregeln für String-Variable.

größe Anzahl der Speicherplätze (im Bereich zwischen Ø und 255). Beachten Sie, wenn Sie eine Größenordnung angeben, daß ein Speicherplatz mehr bereitgestellt wird.

länge (fakultativ) Wenn sie angegeben wird, spezifiziert sie die Länge der einzelnen, im Feld enthaltenen Strings. Die Länge muß im Bereich zwischen 1 und 8Ø liegen. Wird die Länge nicht angegeben, erhalten die Strings automatisch eine Länge von 16 Zeichen.

Beispiele für mögliche DIMensionierungsbefehle für String-Felder sind:

```
DIM X$(4)
DIM NM$(19) *10
DIM IN$(1) * 80
DIM R$(0) *26
```

Im ersten Fall wird ein Feld mit fünf Strings aufgebaut, in denen je 16 Zeichen gespeichert werden können. Der zweite DIM-Befehl baut ein Feld NM auf mit elf Strings zu je 10 Zeichen. Definiert man auch Stringlängen, die kleiner als 16 Zeichen sind, spart man natürlich Speicherkapazität. Das dritte Beispiel zeigt ein Feld IN mit zwei Strings zu je 80 Buchstaben, und im letzten Fall wird ein Feld R mit einem einzigen, 26 Buchstaben umfassenden String aufgebaut.

Neben den einfachen Feldern, die wir gerade behandelt haben, läßt der PC 1350 auch zweidimensionale Felder zu. Ein eindimensionales Feld ist eine Liste von Daten, die in einer einzigen Spalte angeordnet sind. Entsprechend ist ein zweidimensionales Feld eine Tabelle mit Zeilen und Spalten. Zweidimensionale Felder werden folgendermaßen DIMensioniert:

DIM variablenname (zeilen, spalten)

oder

DIM stringvariablenname (zeilen, spalten) \* länge

Dabei bedeutet:

- zeilen                   Gibt die Anzahl der Zeilen im Feld an. Dies muß eine Zahl zwischen 0 und 255 sein. Beachten Sie, daß wenn Sie eine Größenordnung eingegeben haben, eine Zeile mehr bereitgestellt wird.
- spalten                   gibt die Anzahl der Spalten im Feld an. Dies muß eine Zahl zwischen 0 und 255 sein. Beachten Sie, daß, wenn Sie eine Größenordnung eingeben, eine Spalte mehr eingerichtet wird.

Die folgende Tabelle illustriert die Speicherplätze, die sich aus der Anweisung DIM T (2, 3) und den (jetzt aus zwei Zeichen bestehenden) Indizes, die zu den jeweiligen Speicherplätzen gehören, ergibt.

	Spalte 0	Spalte 1	Spalte 2	Spalte 3
Zeile 0	T(0,0)	T(0,1)	T(0,2)	T(0,3)
Zeile 1	T(1,0)	T(1,1)	T(1,2)	T(1,3)
Zeile 2	T(2,0)	T(2,1)	T(2,2)	T(2,3)

**Achtung:** Zweidimensionale Felder nehmen viel Speicherplatz in Anspruch! Z. B. benötigt ein Feld mit 25 Zeilen und 35 Spalten 875 Speicherplätze.

Felder sind sehr nützliche Programmier-Hilfsmittel.

Die folgende Tabelle zeigt die Anzahl der Bytes, die zur Definierung jeder einzelnen Variable benötigt werden sowie die Anzahl der für jeden einzelnen Programm-Befehl erforderlichen.

Variable	Name der Variablen	Daten	
Numerische Variable	7 Bytes	8 Bytes	
String-Variable	7 Bytes	String-Variable	Spezifizierte Anzahl
		einfache Variable (2-Zeichen-Variable)	16 Bytes

Wenn z. B. DIM Z\$(2,3)\*10 spezifiziert wird, werden 12 Variable mit einer Speicherkapazität von je 10 Zeichen bereitgestellt. Dies erfordert 7 Bytes (Name der Variablen) + 10 Bytes (Anzahl der Zeichen) x 12 = 127 Bytes.

Elemente	Zeilennummer	Befehl Funktion	Andere, ENTER
Anzahl der verwendeten Bytes	3 Bytes	1 Byte	1 Byte

#### VARIABLE DER FORM A ( )

Wird im Speicher des Computers ein Bereich für vorgegebene Variable reserviert, kann dieser auch benutzt werden, um mit einem Index versehene Variable (die die gleiche Form wie Feld-Variable haben) zu definieren.



26 vorbestimmte Variablenamen stehen zur Verfügung, nämlich A bis Z (A\$ bis Z\$). Jeder dieser Namen kann als Index eine der Zahlen von 1 bis 26 tragen, wie z. B. A(1) - A(26) oder A\$(1) - A\$(26). D. h., daß die Variable A(1) anstelle der Variablen A benutzt werden kann, A(2) anstelle von B, A(3) anstelle von C usw.

Wenn allerdings schon ein Feld mit dem Namen A oder A\$ durch einen DIM-Befehl definiert wurde, können Index-Variablen mit den Namen A nicht zusätzlich definiert werden. Wurde beispielsweise ein Feld A definiert durch DIM A(5), sind die Speicherplätze A(0) bis A(5) im Programm-/Datenbereich reserviert. Wenn man dann eine Variable A(2) spezifiziert, bezieht sich diese nicht auf die vorbestimmte Variable B, sondern auf die Feld-Variable A(2) im Programm-/Datenbereich. Spezifizieren Sie A(9), wird dies eine Fehlermeldung bewirken, da A(9) außerhalb des durch den DIM A(5)-Befehl bestimmten Bereiches liegt.

Wenn andererseits Index-Variable der Form A( ) eingegeben worden sind, ist es nicht möglich, gleichzeitig Felder A oder A\$ mit Hilfe eines DIM-Befehls einzurichten, es sei denn, die Definition der Index-Variablen wird mit dem CLEAR-Befehl gelöscht.

#### DER GEBRAUCH VON INDIZES GRÖßER ALS 26

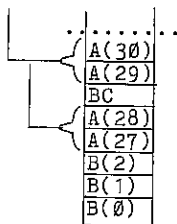
Sollen Indizes größer als 26 für Variable der Form A( ) benutzt werden, wenn ein Feld A nicht durch ein DIM-Statement definiert wurde, werden die entsprechenden Speicherplätze im Programm-/Datenbereich für diese A( )-Variablen definiert. Wenn Sie z. B. A(35) = 5 eingeben, werden die Speicherplätze A(27) bis A(35) im Programm-/Datenbereich reserviert.

Da Variable mit einem Index größer als 26 wie Feld-Variable behandelt werden, unterliegen sie den folgenden besonderen Beschränkungen:

- (1) Die Speicherplätze für ein Feld mit demselben Namen müssen im Programm-/Datenbereich nebeneinander liegen. Andernfalls erhalten Sie eine Fehlermeldung.

```
10 DIM B(2)
20 A(28) = 5
30 BC = 12
40 A(30) = 9
```

Wenn dieses Programm eingegeben wird, befinden sich die Definitionen für das Feld "A" nicht in zwei aufeinanderfolgenden Segmenten des Programm-/Datenbereiches und aus Zeile 40 wird sich ein Fehler ergeben.



- (2) Numerische Feld-Variable und String-Feld-Variable mit demselben Index können nicht zur gleichen Zeit definiert werden. So können z. B. A(30) und A\$(30) nicht zur gleichen Zeit definiert werden, da sie im Programm-/Datenbereich den gleichen Platz einnehmen.
- (3) Zweidimensionale Felder können nicht definiert werden, ebensowenig ist es möglich, die Länge der Zeichenstrings zu definieren, die in einer Feld-Variable enthalten sein sollen. Die Länge des Buchstabenstrings, der von der Variablen A\$( ) erfaßt werden kann, ist beispielsweise auf 7 oder weniger Zeichen beschränkt.
- (4) Variablen mit dem Index (0) können nicht definiert werden. Wenn Sie A(0) oder A\$(0) definieren, werden Sie eine Fehlermeldung erhalten.
- (5) Wenn A(27) oder A\$(27) und größer als erste benutzt werden, werden 7 Bytes für den Variablennamen und 8 Bytes für jede Variable belegt.

## AUSDRÜCKE

Ein Ausdruck ist eine Kombination von Variablen, Konstanten und Operatoren, die durch Rechenoperation auf einen einzigen Wert zusammengefaßt werden kann. Die Rechenbeispiele, die Sie in Kapitel 3 eingegeben haben, waren Beispiele für solche Ausdrücke. Ausdrücke sind ein wesentlicher Bestandteil von BASIC-Programmen. Z. B. kann ein Ausdruck eine Formel sein, mit der das Ergebnis einer Gleichung ermittelt wird, oder ein Test zur Bestimmung des Verhältnisses zwischen zwei Größen, ein Mittel, um eine Reihe von Strings zu formatieren.

## NUMERISCHE OPERATIONEN

Der PC 1350 verfügt über fünf numerische Operatoren. Dies sind die arithmetischen Rechenzeichen, die Sie benutzt haben, als Sie die Benutzung des PC 1350 als Taschenrechner in Kapitel 3 kennenlernten.

- + Addition
- Subtraktion
- \* Multiplikation
- / Division
- ^ Exponentiation

Ein numerischer Ausdruck wird in derselben Weise konstruiert, wie Sie komplexe Rechenbefehle eingegeben haben. Numerische Ausdrücke können jede aussagefähige Kombination von numerischen Konstanten, numerischen Variablen und numerischen Operatoren beinhalten:

```
(A*B)^2
A(2,3)4A(3,4)45,0-C
(A/B)*(C+D)
```

#### STRING-AUSDRÜCKE

String-Ausdrücke sind ähnlich den numerischen Ausdrücken, allerdings gibt es nur einen einzigen String-Operator: die Verkettung (+). Das benutzte Symbol ist dasselbe wie das Plus-Zeichen. Wird es mit einem String-Paar benutzt, knüpft das + den zweiten String an das Ende des ersten an und schafft dadurch einen längeren String. Wenn Sie komplexere String-Verkettungen und andere String-Operationen vornehmen, bedenken Sie bitte, daß der PC 1350 nur 80 Zeichen zur Zeit annimmt.

**Achtung:** String-Einheiten und numerische Einheiten können nicht in demselben Ausdruck definiert werden, es sei denn, man benutzt eine der Funktionen, die String-Werte in numerische Werte umwandeln oder umgekehrt.

```
"15" + 10      ist unzulässig
"15" + "10"    ist "1510", nicht "25"
```

#### VERHÄLTNIS-AUSDRÜCKE

Ein Verhältnis-Ausdruck vergleicht zwei Ausdrücke und gibt an, ob das festgestellte Verhältnis wahr oder unwahr ist. Die Verhältnis-Operatoren sind:

```
>   größer als
>=  größer oder gleich
=   gleich
<>  ungleich
<=  kleiner oder gleich
<   kleiner als
```

Die folgenden Ausdrücke werden als Verhältnis-Ausdrücke akzeptiert:

```
A<B
C(1,2)>=5
D(3)<>8
```

Wenn A gleich 10 wäre, B gleich 12, C(1,2) gleich 6 und D(3) gleich 9, wären alle diese Ausdrücke wahr.

Zeichen-Strings können ebenfalls mit Hilfe von Verhältnis-Ausdrücken verglichen werden. Die beiden Strings werden Zeichen für Zeichen gemäß dem Wert ihres ASCII-Codes verglichen (siehe Anhang B: ASCII-Code-Tabelle). Ist ein String kürzer als der andere, wird  $\emptyset$  oder NUL in die freibleibenden Positionen eingesetzt. Alle folgenden Beispiele sind wahr:

```
"ABCDEF" = "ABCDEF"
"ABCDEF" <> "ABCDE"
"ABCDEF" > "ABCDE"
```

Verhältnis-Ausdrücke beurteilen nach wahr oder unwahr. Beim PC 1350 wird "wahr" durch eine 1 angegeben, "unwahr" durch eine 0. In jeder logischen Überprüfung wird ein Ausdruck, der 1 oder mehr ergibt, als wahr betrachtet, während jeder, der 0 ergibt, als unwahr verstanden wird.

#### LOGISCHE AUSDRÜCKE

Logische Ausdrücke sind Verhältnis-Ausdrücke, die die Operatoren AND, OR und NOT benutzen. AND und OR werden verwendet, um zwei Verhältnis-Ausdrücke miteinander zu verbinden, der Wert des verbundenen Ausdrucks wird aus der folgenden Tabelle ersichtlich:

A AND B

Wert von A

		wahr	unwahr
Wert von B	wahr	wahr	unwahr
	unwahr	unwahr	unwahr

A OR B

Wert von A

		wahr	unwahr
Wert von B	wahr	wahr	wahr
	unwahr	wahr	unwahr

Dezimalzahlen können folgendermaßen in binärer Schreibweise von 16 Bit dargestellt werden:

Dezimalform	Binärform
32767	0111111111111111
.	.
.	.
.	.
3	0000000000000011
2	0000000000000010
1	0000000000000001
0	0000000000000000
-1	1111111111111111
-2	1111111111111110
-3	1111111111111101
.	.
.	.
.	.
-32767	1000000000000000

Die negative Form einer Dezimalzahl wird folgendermaßen dargestellt.

NOT	0000000000000001
(Negativ)	1111111111111110

So wird also für jedes Bit 1 zu 0 interpretiert, und 0 zu 1. Dies nennt man "Inversion(NOT)".

Demnach ergibt sich das Folgende, wenn man 1 und NOT 1 addiert:

$$\begin{array}{r}
 0000000000000001 \quad (1) \\
 (+) \quad 1111111111111110 \quad (\text{NOT } 1) \\
 \hline
 1111111111111111 \quad (-1)
 \end{array}$$

Alle Bits werden also 1. Entsprechend der obigen Zahlenliste werden die Bits in dezimaler Schreibweise -1, d. h. 1 + NOT 1 = -1.

Das Verhältnis zwischen einem numerischen Wert X und seiner negativen Form (NOT X) ist:

$$X + \text{NOT } X = -1$$

Daraus ergibt sich die Gleichung NOT X = -X-1, d. h. NOT X = -(X+1).

Aus dieser Gleichung lassen sich folgende Ergebnisse ableiten:

$$\begin{array}{l}
 \text{NOT } 0 = -1 \\
 \text{NOT } -1 = 0 \\
 \text{NOT } -2 = 1
 \end{array}$$

Mit diesen logischen Operationen können Sie auch mehr als zwei Ausdrücke in Beziehung setzen. Dann sollten Sie aber darauf achten, Klammern zu verwenden, um den gewünschten Vergleich zu verdeutlichen.

(A<9) AND (B>5)  
(C=5) OR (C=6) OR (C=7)

Der PC 1350 behandelt logische Operationen "bitweise" als logische Funktionen auf der Basis von 16 Bit (siehe Anmerkung zu den Verhältnis-Ausdrücken und wahr/unwahr). In normalen Operationen hat dies keine Bedeutung, weil die einfache 1 oder 0 (wahr oder unwahr), die aus Verhältnis-Ausdrücken resultieren, nur je ein Bit einnehmen. Wenn Sie jedoch logische Operatoren auf andere Werte als 0 oder 1 anwenden, wird jedes Bit einzeln bearbeitet. Wenn z. B. A = 17 ist, und B = 22, dann ist (A OR B) = 23.

17 ist in binärer Schreibweise 10001  
22 ist in binärer Schreibweise 10110  
17 OR 22 ist 10111  
10111 in dezimaler Schreibweise ist aber 23.

Wenn Sie ein geübter Programmierer sind, kennen Sie sicherlich Anwendungen, wo diese Art der Operation sehr nützlich sein kann. Programmieranfänger sollten sich vorerst an klare, einfache Wahr-/Unwahr-Aussagen halten.

#### KLAMMERUNG UND VORRANG DER OPERATOREN

Bei der Bearbeitung komplexer Ausdrücke folgt der PC 1350 einer Reihe vordefinierter Prioritäten, die bestimmen, in welcher Reihenfolge die Operatoren bearbeitet werden. Das kann sehr wichtig sein:

5 + 2 \* 3 kann sein:

5 + 2 = 7           oder           2 \* 3 = 6  
7 \* 3 = 21                           6 + 5 = 11

Die genauen Regeln des "Operatoren-Vorrangs" werden in Anhang D beschrieben.

Damit Sie sich nicht alle diese Regeln merken müssen, und damit Sie Ihre Programme eindeutiger gestalten, benutzen Sie immer Klammern, um die Reihenfolge der Bearbeitung von Operatoren vorzugeben. Das obige Beispiel wird eindeutig, wenn Sie schreiben:

(5 + 2) \* 3           oder   5 + (2 \* 3)

#### RUN-MODUS

Im allgemeinen kann jeder der oben angesprochenen Ausdrücke ebenso im RUN-Modus benutzt werden wie bei der Programmierung eines BASIC-Statements. Im RUN-Modus wird ein Ausdruck sofort ausgewertet und angezeigt, z. B.

#### INGABE

(5>3) AND (2<6)

#### ANZEIGE

(5>3) AND (2<6)

Die 1 gibt an, daß der Ausdruck wahr ist.

## FUNKTIONEN

Funktionen sind spezielle Bestandteile der BASIC-Sprache, die einen Wert in einen anderen Wert transformieren. Funktionen arbeiten wie Variable, deren Wert durch andere Variable oder Ausdrücke festgelegt wurde. ABS ist eine Funktion, die den absoluten Wert der Zahl hervorbringt, auf die sie angewendet wird.

ABS (-5) ist 5  
ABS (6) ist 6

LOG ist eine Funktion, die den dekadischen Logarithmus der Zahl berechnet, auf die LOG angewendet wird.

LOG (100) ist 2.  
LOG (1000) ist 3.

Eine Funktion kann überall dort verwendet werden, wo auch Variable benutzt werden können. Viele Funktionen erfordern keine Klammerung.

LOG 100 ist dasselbe wie LOG (100)

Sie müssen jedoch Klammern benutzen, sobald sich eine Funktion auf mehr als eine Zahl bezieht. Der Gebrauch von Klammern sorgt immer für eine klarere und eindeutigere Programmierung.

In Kapitel 9 finden Sie eine vollständige Liste der auf dem PC 1350 verfügbaren Funktionen.

## KAPITEL 5

## PROGRAMMIEREN MIT DEM PC 1350

Im vergangenen Kapitel haben wir einige Begriffe und Ausdrücke der Programmiersprache BASIC kennengelernt. In diesem Kapitel wollen wir nun diese Elemente benutzen, um auf dem PC 1350 Programme zu schreiben. Wir möchten noch einmal darauf hinweisen, daß dieses Handbuch nicht als Einführung in die BASIC-Programmierung verstanden werden soll. Dieses Kapitel soll Sie lediglich in den besonderen Gebrauch des BASIC auf dem PC 1350 einführen.

## PROGRAMME

Ein Programm besteht aus einer Reihe von an den Computer gerichteten Befehlen. Denken Sie daran, daß der PC 1350 nur eine Maschine ist, die genau die Operationen durchführt, die Sie angeben. Sie als Programmierer sind dafür verantwortlich, korrekte Befehle einzugeben.

## BASIC STATEMENTS

Der PC 1350 setzt Programme entsprechend einem bestimmten Format um. Dieses Format wird Statement genannt. Sie geben die BASIC-Statements immer nach einem bestimmten Muster ein. Ein Statement muß mit einer Zeilennummer beginnen.

```
10: PRINT "HALLO"
```

## ZEILENUMMERN

Jede Programmzeile muß eine eigene Nummer haben, und zwar muß dies eine ganze Zahl zwischen 1 und 65279 sein. Zeilennummern sind die Bezugspunkte des Computers. Sie geben dem PC 1350 an, in welcher Reihenfolge er ein Programm abarbeiten muß. Es ist nicht erforderlich, daß Sie die Programmzeilen folgerichtig eingeben (obwohl dieser Weg sicher der weniger verwirrende ist - besonders, wenn Sie noch Anfänger sind). Der Computer beginnt beim Durcharbeiten eines Programmes immer mit der niedrigsten Zeilennummer und arbeitet die folgenden in aufsteigender Reihenfolge ab.

Beim Programmieren ist es sinnvoll, genug Raum für spätere Einschübe zwischen den einzelnen Zeilen zu lassen (10, 20, 30, ...10, 30, 50 etc.).

Benutzen Sie keine gleichen Zeilennummern in verschiedenen Programmen: Wenn Sie gleiche Zeilennummern benutzen, wird die ältere Zeile gelöscht, sobald Sie eine neue mit derselben Nummer eingeben.

## BASIC-BEFEHLE

Alle BASIC-Statements müssen Befehle enthalten. Diese Befehle sagen dem Computer, welche Operationen er durchführen soll. Ein Befehl ist immer Programmbestandteil, insofern wird die Operation nicht direkt darauf erfolgen.



Einige Statements erfordern oder erlauben den Gebrauch von Operanden:

```
10: PRINT "HALLO"
20: READ B(10)
30: END
```

Operanden informieren den Computer darüber, auf welche Daten sich die vom \*Befehl angeordnete Operation bezieht. Einige Befehle erfordern Operanden, im Zusammenhang mit anderen Befehlen können (müssen aber nicht) Operanden eingesetzt werden.

(In Kapitel 9 finden Sie eine vollständige Übersicht über die auf PC 1350 möglichen BASIC-Befehle und ihren Gebrauch.)

Beachten Sie: Befehle, Kommandos und Funktionen müssen in Großbuchstaben eingegeben werden.

### BASIC-KOMMANDOS

Kommandos sind Anweisungen an den Computer, die außerhalb des Programms eingegeben werden. Kommandos fordern den Computer auf, bestimmte Handlungen mit dem Programm vorzunehmen oder einen bestimmten Modus zu setzen, der dann wiederum die Art der Programmabarbeitung determiniert.

Anders als die Befehle haben Kommandos unmittelbare Wirkung - sobald Sie die Eingabe des Kommandos beendet haben (mit Betätigen der **ENTER**-Taste), wird das Kommando ausgeführt. Kommandos geht keine Zeilennummer voraus.

```
RUN
NEW
RADIAN
```

Einige Kommandos können auch als Befehle verwendet werden. (In Kapitel 9 findet sich eine vollständige Übersicht der Kommandos und ihres Gebrauchs auf dem PC 1350.)

### MODI

Sie erinnern sich sicherlich, daß Sie, als Sie den PC 1350 als Rechner ohne Programmunterstützung benutzt haben, den RUN-Modus gewählt haben.

Der RUN-Modus wird ebenfalls gebraucht, um die von Ihnen geschriebenen Programme abzarbeiten.

Der PROGRAMM-Modus wird gewählt, wenn Sie Programme eingeben oder editieren wollen.

Der ReSeRve-Modus ermöglicht es, vordefinierte String-Variable zu kennzeichnen und zu speichern. Weiterhin wird er für komplexere Programmierung gebraucht (siehe Kapitel 6).

## DER ANFANG DES PROGRAMMIERENS MIT DEM PC 1350

Nachdem Sie in vielen Übungen den PC 1350 als Taschenrechner benutzt haben, sind Sie sicherlich schon recht vertraut mit der Tastatur. Von nun an werden wir, wenn wir eine Eingabe zeigen, nicht mehr jede Tastenbedienung zeigen. Denken Sie immer daran, **SHIFT** zu betätigen, um Zugang zu den Zeichen oberhalb einzelner Tasten zu bekommen. Vergessen Sie nicht, AM ENDE JEDER ZEILE DIE **ENTER** -TASTE ZU DRÜCKEN!

Nun können Sie mit dem Programmieren beginnen.

Um Programmeingaben machen zu können, schalten Sie den Computer erst in den PROGRAMM-Modus, indem Sie die Taste **MODE** benutzen. Auf dem Display wird folgende Anzeige erscheinen:

EINGABEANZEIGE**MODE**

```
PROGRAM MODE
>
```

Geben Sie das **NEW** -Kommando ein.

EINGABEANZEIGE**NEW**

```
NEW
>
```

Das **NEW**-Kommando löscht aus dem Arbeitsspeicher des PC 1350 alle vorhandenen Daten und Programme. Das Bereitschaftssymbol erscheint, wenn Sie die **ENTER** -Taste gedrückt haben. Es zeigt an, daß der Computer nun Eingaben erwartet.

Beispiel 1:

Vergewissern Sie sich, daß der PC 1350 sich im PRO-Modus befindet, und geben Sie das folgende Programm ein:

EINGABEANZEIGE**10 PRINT "HALLO"**

```
10: PRINT "HALLO"
```

Beachten Sie, daß der PC 1350 ihre Eingabe anzeigt, sobald Sie **ENTER** gedrückt haben, und daß er dabei einen Doppelpunkt (:) zwischen die Zeilennummer und den Befehl einfügt. Vergewissern Sie sich, daß das Statement in der richtigen Form eingegeben wurde.

Nun schalten Sie den Computer auf den RUN-Modus um:

EINGABEANZEIGE

RUN

RUN HALLO
--------------

Da dies die einzige Programmzeile ist, wird der Computer an dieser Stelle stoppen. Drücken Sie **ENTER**, um aus dem Programm herauszukommen und geben Sie RUN vor, wenn das Programm noch einmal ablaufen soll.

## Beispiel 2:

## DAS EDITIEREN EINES PROGRAMMS

Nehmen wir an, Sie wollen den Inhalt dessen, was Ihr Programm anzeigt, verändern, d. h. Sie wollen Ihr Programm editieren. Bei einem einzeiligen Programm könnten Sie die Eingabe schlicht wiederholen, aber wenn Sie komplexere Programme schreiben, wird das Editieren zu einem wichtigen Bestandteil des Programmierens. Wir wollen das Programm, das Sie gerade geschrieben haben, editieren.

Sind Sie noch im RUN-Modus? Wenn ja, betätigen Sie die **MODE**-Taste, um wieder in den PROGRAMM-Modus zu gelangen.

Um Ihr Programm editieren zu können, müssen Sie es nun wieder abrufen. Benutzen Sie dazu die **↑**-Taste. Sofern Ihr Programm vollständig durchlaufen worden war, wird die **↑**-Taste lediglich die letzte Programmzeile abrufen. Falls ein Fehler im Programm vorlag oder Sie die Ausführung mit **BRK** unterbrochen hatten, wird die **↑**-Taste die Zeile abrufen, in der der Fehler liegt oder in der **BRK** betätigt wurde. Um Veränderungen im Programm vorzunehmen, bedienen Sie sich der **↑**-Taste, um sich im Programm nach oben zu bewegen (d. h. die jeweils vorherige Zeile abzurufen), der **↓**-Taste, um sich nach unten zu bewegen (d. h. die jeweils nächste Zeile abzurufen). Wenn die **↑**- und **↓**-Tasten festgehalten werden, "rollt" das Programm im Display nach oben bzw. unten, d. h. die einzelnen Zeilen werden nacheinander durch das Display laufen.

Sie erinnern sich sicherlich, daß Sie den Cursor innerhalb einer Programmzeile (die jeweils in der obersten Zeile des Displays angezeigt wird) mit den **◀**- und **▶**-Tasten bewegen können. Mit Hilfe des **▶**-Pfeiles setzen Sie nun den Cursor auf das erste Zeichen, das Sie ändern möchten.

Beachten Sie: Obwohl auf dem Display mehrere Zeilen angezeigt werden können, kann der Cursor nur in der obersten Zeile bewegt werden. Um eine andere Zeile zu editieren, müssen Sie die gewünschte Zeile nach oben verschieben. Dazu benutzen Sie die **↓**-Taste und können dann editieren.

EINGABE

↑

◀◀◀◀◀

ANZEIGE

10: PRINT "HALLO"

10: PRINT "HALLO"

Beachten Sie, daß der Cursor die Form eines blinkenden Rechtecks angenommen hat, um so anzuzeigen, daß er auf einem schon bestehenden Zeichen steht.

EINGABE

TSCHUSS!"

ANZEIGE

10: PRINT "TSCHUSS!"

Vergessen Sie nicht, am Ende der Zeile **ENTER** zu drücken. Schalten Sie um in den RUN-Modus.

EINGABE

RUN

ANZEIGERUN  
ERROR 1 IN 10

Dies ist ein neuer Typ von Fehlermeldung. Der Fehler ist nicht nur identifiziert (unser alter Freund, der Syntax-Fehler), sondern auch die Nummer der Zeile, in der er auftritt, wird angegeben.

Drücken Sie **CLS**, um die Fehlermeldung zu beseitigen. Schalten Sie zurück in den PROGRAM-Modus. Sie müssen unbedingt im PROGRAM-Modus sein, um Änderungen im Programm vornehmen zu können. Mit Hilfe des **↑**-Pfeiles rufen Sie nun die letzte Zeile Ihres Programms ab.

EINGABE

↑

ANZEIGE

10: PRINT "TSCHUSS"

Der blinkende Cursor steht auf der kritischen Stelle. In Kapitel 4 haben Sie gelernt, daß Sie, wenn Sie String-Konstanten eingeben, alle dazugehörigen Zeichen in Anführungszeichen einschließen müssen. Benutzen Sie die **DEL**-Taste, um das Ausrufungszeichen zu löschen.

EINGABE**DEL**ANZEIGE

10 PRINT "TSCHUSS"

Nun wollen wir das ! an die richtige Stelle setzen. Wenn man Programme editiert, werden die DEL- und INS-Funktionen genauso benutzt wie beim Korrigieren einfacher Rechenoperationen ohne Programmunterstützung (siehe Kapitel 3). Mit Hilfe des **◀**-Pfeiles bringen Sie den Cursor auf das erste der Einfügung folgende Zeichen.

EINGABEANZEIGE

```
10 PRINT "TSCHUESS"
```

Drücken Sie die **INS**-Taste. Ein "□" markiert die Stelle, an der ein neues Zeichen einzugeben ist.

EINGABE

**INS**

ANZEIGE

```
10 PRINT "TSCHUESS□"
```

Geben Sie das ! ein. Die Anzeige sieht nun folgendermaßen aus:

EINGABE

!

ANZEIGE

```
10 PRINT "TSCHUESS!"
```

Denken Sie daran, **ENTER** zu drücken, damit die Korrektur in das Programm aufgenommen wird!

**Achtung:** Wenn Sie eine ganze Zeile aus Ihrem Programm löschen möchten, geben Sie nur die Zeilennummer ein und die ursprüngliche Zeile wird gelöscht.

Werden im PROGRAMM-Modus Daten eingegeben, ohne daß der Cursor angezeigt ist, werden die entsprechenden Zeichen gewöhnlich in der linken Spalte der obersten Zeile der Anzeige angezeigt. Benutzen Sie hingegen die Pfeiltasten, wenn das Bereitschaftssymbol angezeigt wird, werden Ihre Eingaben sukzessive anstelle des Bereitschaftssymbols angezeigt.

**Beispiel 3:****DER GEBRAUCH VON VARIABLEN BEIM PROGRAMMIEREN**

Wenn Ihnen der Gebrauch von numerischen und String-Variablen nicht vertraut ist, lesen Sie noch einmal die entsprechenden Abschnitte in Kapitel 4.

Der Gebrauch von Variablen beim Programmieren erlaubt eine sehr differenzierte Ausnutzung der Fähigkeiten Ihres PC 1350.


Erinnern Sie sich, daß Sie einfache numerische Variable mit Buchstaben von A bis Z belegen.

A = 5

Um String-Variablen zu bestimmen, benutzen Sie ebenfalls einen Buchstaben, gefolgt von einem Dollarzeichen. Benutzen Sie niemals den gleichen Buchstaben, um eine numerische und eine String-Variablen zu bestimmen. Sie können nicht A und A\$ im selben Programm benutzen.

Erinnern Sie sich, daß einfache String-Variable nicht mehr als 7 Zeichen umfassen können.




A\$ = "TOTAL"

Anmerkung: Wenn während der Programmausführung ein Fehler auftritt und dieser am Ende einer Zeile erscheint, benutzen Sie die -Taste, bevor Sie an dieser Stelle Text einfügen.

BEISPIEL:

EINGABE

ANZEIGE



| ERROR 1 IN 20



| 20:S=π \*R\*\_

↑  
Cursor-Position

 R 

| 20:S=π \*R\*R

Wenn Sie die -Taste nicht benutzen, kann der Anzeigehalt unter Umständen vom Bildschirm verschwinden, wenn Sie weiteren Text hinzufügen. Indem Sie die Taste  drücken, erscheint der angezeigte Text einschließlich des hinzugefügten.

Anmerkung: Fehler und Unterbrechungen während der Programmausführung werden in der Zeile angezeigt, die der manuellen Operation oder dem PRINT-Kommando folgt. Der Fehler oder die Unterbrechung wird nicht angezeigt bei Fehlern oder Unterbrechungen (Ausführung eines STOP-Befehls oder einer zeitweisen Unterbrechung, während die -Taste gedrückt wird), wenn die Display-Anfangsposition durch das CURSOR-Kommando definiert wurde. Der Rechner ist dann in jedem Fall im Fehler- oder Unterbrechungs-Modus. Im Fehler-Modus können Sie die alphabetischen und numerischen Tasten nicht benutzen. Beseitigen Sie den Fehler, indem Sie die Taste  drücken.

Die einer Variablen zugewiesenen Werte können sich im Laufe eines Programms ändern, sie können jeweils den im Programm eingegebenen oder errechneten Wert annehmen. Eine Möglichkeit der Variablenzuweisung ist, den Befehl INPUT zu benutzen. Im folgenden Programmbeispiel ändert sich der Wert der Variablen A\$ aufgrund der Daten, die eingegeben werden, um die Frage "WORT?" zu beantworten. Geben Sie dieses Programm ein:

```
10 INPUT "WORT?";A$
20 B= LEN (A$)
30 PRINT "WORT_HAT";B;"_BUCHSTABEN"
40 END
```

} bedeutet: Leerschritt

Da Zeile 30 des Programms länger als 24 Zeichen ist, wird der verbleibende Teil in der folgenden Zeile angezeigt.

Das zweite neue Element in diesem Programm ist der Gebrauch des END-Statements, um die Beendigung des Programms anzuzeigen. END läßt den Computer wissen, daß das Programm beendet ist. Es ist eine gute Übung, beim Programmieren immer das END-Statement zu benutzen.

Wenn Ihre Programme komplexer werden, möchten Sie sie möglicherweise noch einmal durchsehen, bevor Sie sie abarbeiten lassen. Zu diesem Zweck benutzen Sie das LIST-Kommando. LIST, das nur im PRO-Modus benutzt werden kann, bringt das gesamte Programm - angefangen bei der niedrigsten Zeilennummer - zur Anzeige.

Versuchen Sie, dieses Programm auflisten zu lassen.

EINGABEANZEIGE

LIST

```
10: INPUT "WORT?";A$
20: B= LEN (A$)
30: PRINT "WORT HAT ";B;"B
    UCHSTABEN"
```

Benutzen Sie die  $\uparrow$ - und  $\downarrow$ -Pfeile, um sich in Ihrem Programm zu bewegen, bis Sie das gesamte Programm durchgesehen haben. Um eine Zeile noch einmal anzusehen, die mehr als 24 Zeichen enthält, bringen Sie den Cursor an die äußerste rechte Seite des Displays und die weiteren Zeichen werden in der Anzeige erscheinen. Nachdem Sie Ihr Programm kontrolliert haben, lassen Sie es ablaufen:

EINGABEANZEIGE

RUN

```
RUN
WORT?_
```

HILFE

```
RUN
WORT?HILFE_
WORT HAT 5. BUCHSTABEN
```

ENTER

```
RUN
WORT?HILFE_
WORT HAT 5. BUCHSTABEN
>
```

Dies ist das Ende Ihres Programms. Natürlich können Sie es noch einmal ablaufen lassen, indem Sie RUN eingeben. Auf jeden Fall wäre das Programm unterhaltsamer, wenn es mehr als eine Eingabemöglichkeit enthielte. Wir wollen das Programm nun so modifizieren, daß es weiterläuft, ohne daß man nach jeder Antwort RUN eingeben muß.

Gehen Sie zurück in den PRO-Modus und benutzen Sie die Auf- und Abwärts-Pfeile (oder LIST), um in Zeile 40 zu kommen.

EINGABEANZEIGELIST 40 

Sie können nun 40 eingeben, um so die gesamte Zeile zu löschen. Sie können aber auch den Cursor auf das E in END setzen. Ändern Sie die Zeile 40 folgendermaßen:

40: GOTO 10

Nun lassen Sie das modifizierte Programm ablaufen.

Das GOTO-Statement schafft eine Programm-Schleife (d. h. eine bestimmte Operation wird immer wieder durchgeführt). Da Sie der Schleife kein Ende gesetzt haben, wird sie sich nicht unterbrechen. Man spricht hier von einer Endlos-Schleife. Um dieses Programm zu unterbrechen, bedienen Sie die  -Taste.

Wenn Sie das Programm mit der  -Taste unterbrochen haben, können Sie es mit dem CONT-Kommando neu starten. CONT steht für CONTINUE (fortsetzen). Bei Eingabe des CONT-Kommandos läuft das Programm von da an weiter, wo es mit der  -Taste unterbrochen wurde.

#### Beispiel 4: KOMPLEXERE PROGRAMME

Das folgende Programm berechnet N-Fakultät (N!). Das Programm beginnt mit 1 und berechnet N! bis zu der Grenze, die Sie eingeben. Geben Sie das Programm ein:

```
100 F=1: WAIT 128
110 INPUT "GRENZE? ";L
120 FOR N=1 TO L
130 F=F*N
140 PRINT N,F
150 NEXT N
160 END
```

In diesem Programm sind einige neue Erscheinungen enthalten. Der Befehl WAIT in Zeile 100 kontrolliert den Zeitraum, über den eine Zeile im Display bleibt, ehe das Programm fortgesetzt wird. Die Zahlen und ihre Fakultäten werden angezeigt, wenn sie berechnet werden. Durch das WAIT-Statement bleiben sie etwa 2 Sekunden in der Anzeige, statt darauf zu warten, daß Sie  drücken.

Ebenfalls in Zeile 100 tauchen zwei Statements auf, die durch einen Doppelpunkt getrennt werden. Sie können so viele Statements in eine Zeile schreiben, wie Sie möchten, solange Sie die einzelnen Statements durch Doppelpunkte voneinander trennen und die maximale Anzahl von 80 Zeichen pro Zeile (einschließlich ) nicht überschreiten. Solche Mehrfach-Statements machen allerdings das Lesen und Korrigieren eines Programms schwieriger, deshalb ist es eine gute Übung, sie nur zu benutzen, wenn die Einzel-Statements sehr kurz und einfach sind oder wenn es einen besonderen Grund dafür gibt, Mehrfach-Statements zu verwenden.



In unserem Programm haben wir in Zeile 120 den Befehl FOR und in Zeile 150 den Befehl NEXT benutzt, um eine Schleife aufzubauen. In Beispiel 3 hatten Sie eine Endlos-Schleife geschaffen, die sich kontinuierlich wiederholte. Mit dieser FOR/NEXT-Schleife addiert der PC 1350 jedes Mal 1 zu N hinzu, wenn der Programmablauf den Befehl NEXT erreicht. Hierbei wird geprüft, ob N die anfangs gesetzte Grenze überschreitet. Ist N kleiner oder gleich der Grenze, beginnt der Programmdurchlauf wieder am Anfang der Schleife. Ist N größer als die Grenze, springt das Programm in Zeile 160 und wird dort beendet.

In einer FOR/NEXT-Schleife können Sie jede numerische Variable benutzen. Sie müssen auch nicht mit 1 anfangen, und Sie können bei jedem Programmschritt jeden beliebigen Betrag dazu addieren. Näheres finden Sie in Kapitel 9.

Wir haben dieses Programm mit Zeilennummern von 100 aufwärts belegt. Eine solche Belegung mit unterschiedlichen Zeilennummern macht es möglich, daß man mehrere Programme zur gleichen Zeit im Arbeitsspeicher hat. Um dieses Programm durchlaufen zu lassen, geben Sie statt RUN 10 ein:

```
RUN 100
```

Außer der Möglichkeit, Programme durchlaufen zu lassen, indem Sie die erste Zeilennummer eingeben, können Sie den Programmen auch einen Buchstaben als Namen geben und sie mit der **DEF**-Taste starten (siehe Kapitel 6).

## DAS SPEICHERN VON PROGRAMMEN

Sie erinnern sich sicherlich, daß Einstellungen, ReSerVe-Funktionen und andere Daten und Funktionen im Speicher Ihres PC 1350 erhalten bleiben, auch wenn Sie ihn abschalten. Programme bleiben ebenfalls erhalten, wenn Sie den Computer abschalten oder er das (durch die AUTO-OFF-Funktion) selbst tut. Selbst, wenn Sie die **BRK** -, **CLS** - oder **CA** -Tasten benutzen, werden die Programme nicht gelöscht.

Programme gehen nur aus dem Speicher verloren, wenn Sie einen der folgenden Abläufe ausführen:

- NEW eingeben, ehe Sie mit dem Programmieren beginnen;
- den Computer mit dem ALL-RESET-Schalter initialisieren;
- ein neues Programm mit denselben Zeilennummern schreiben, die Sie bereits an ein im Speicher vorhandenes vergeben haben;
- die Batterien wechseln.

Diese kurze Einführung in das Programmieren mit dem PC 1350 sollte dazu dienen, die aufregenden Möglichkeiten des Programmierens mit Ihrem neuen Computer näher zu beschreiben. Weitere Übungen finden Sie in Kapitel 9.

## GRAPHIK-FUNKTIONEN

Der Bildschirm des PC 1350 setzt sich aus 32 Zeilen mit 150 Punkten zusammen. Mit speziellen Befehlen kann man mit diesen Punkten einfache Graphiken erstellen.

Die folgenden sechs Befehle werden dafür bereitgestellt:

- GPRINT: Graphic PRINT. Dieses Kommando erzeugt Formen mit 8 vertikalen Punkten pro Einheit.
- GCURSOR: Graphic CURSOR. Dieses Kommando setzt den Cursor, um mit dem Kommando GPRINT Bilder zu erstellen.
- PSET: Point SET. Setzt einen Punkt an der angegebenen Stelle.
- PRESET: Point RESET. Löscht einen Punkt an der angegebenen Stelle.
- LINE: Mit diesem Kommando wird zwischen zwei definierten Punkten eine Linie oder ein Quadrat erzeugt.
- POINT: Dieses Kommando ermöglicht die Abfrage, ob ein Punkt gesetzt ist oder nicht.

Hier soll nur das Konzept der Erstellung von Graphiken erläutert werden. Genauere Informationen über die Befehle finden Sie unter der entsprechenden Beschreibung des Kommandos.

Es gibt zwei Grundversionen, ein Bild zu erzeugen:

Einmal kann man ein Bild aus vordefinierten Symbolen zusammenstellen, die zweite Möglichkeit ist, das Bild aus einzelnen Punkten zu bilden, die man hell definiert oder dunkel läßt.

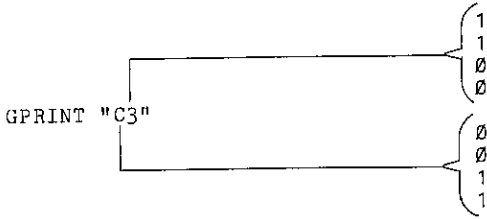
Um eine Graphik mit alphanumerischen Zeichen auf dem Bildschirm zu kombinieren, muß das Graphik-Kommando nach den normalen PRINT-Befehlen gegeben werden.

## (1) Bilder mit vordefinierten Symbolen

Mit dem GPRINT-Kommando und den 16 vorgesehenen Symbolen soll ein Bild erstellt werden. Folgende Symbole stehen zur Verfügung:

Hex-Zahl	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Form	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Jedes dieser Symbole setzt sich aus vier vertikalen Punkten zusammen, eine "1" steht hier für hell, eine "0" für dunkel. Das GPRINT-Kommando setzt nun jeweils zwei Symbole zu einem 8-Punkt-Zeichen zusammen, z. B.:



Die erste Hälfte eines Paares wird also immer für die obere Hälfte des Bildes, die zweite Hälfte des Paares für die untere des Bildes benutzt. Werden mehrere Paare in einem Kommando eingegeben, erhält man umfangreiche Bilder, z. B.



wobei das erste Drittel aus C3, das zweite aus 24 und das letzte aus 18 entstanden ist.

Die einzelnen Paare werden also hintereinander gesetzt.

(2) Die Position auf dem Bildschirm

Wie beschrieben kann ein Bild mit den vordefinierten Symbolen und dem GPRINT-Kommando erstellt werden. Mit dem Kommando GCURSOR kann man die Position des Bildes bestimmen.

Der Bildschirm des PC 1350 wird aus 32 Zeilen mit 150 Punkten gebildet.

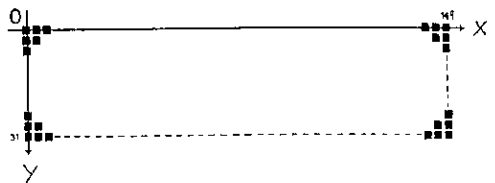
Die Position jedes Punktes kann, wie in einem Koordinatensystem, mit (x,y) ermittelt werden. "x" ist dabei die horizontale, "y" die vertikale Achse. Zu beachten ist, daß ein Punkt umso tiefer auf dem Bildschirm ist, je größer der Wert von "y" ist, also genau entgegen dem normalen Koordinatensystem.

Die Wertebereiche sind für

"x" 0 - 149

und für

"y" 0 - 31



**Achtung:** Die Werte für "x" und "y" müssen im Bereich von -32768 und +32767 liegen. Alle Werte kleiner null oder größer den angegebenen Obergrenzen (149 bzw. 31) können aber nicht auf dem Bildschirm dargestellt werden.

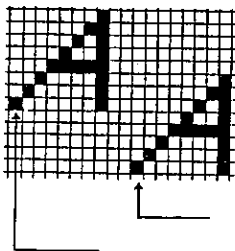
Diese Einschränkung gilt auch für die anderen Graphik-Kommandos, wie z. B. SET, RESET und LINE. Auch hier können nur die Koordinatenwerte von 0 - 149 und 0 - 31 dargestellt werden.

Das GPRINT-Kommando zeichnet Bilder, indem je zwei 4-Punkt-Symbole ausgegeben werden. Das GCURSOR-Kommando gibt den Positionspunkt an. Das gewünschte Zeichen wird nun über diesen Punkt gesetzt, so daß der letzte Punkt des Symbols der Positionspunkt ist.

Beispiel:

```
5 CLS
10 AA$="80402010181412FF"
20 GCURSOR (70,15)
30 GPRINT AA$
40 GCURSOR (80,20)
50 GPRINT AA$
```

Wird das Programm gestartet, werden zwei Bilder ausgegeben. Die jeweils linke, untere Ecke ist die mit GCURSOR vorgegebene Position, also für das erste Bild Spalte 70, Zeile 15, und für das zweite Spalte 80 und Zeile 20.



(3) Ein Bild aus einzelnen Punkten

Die zweite Möglichkeit, ein Bild zu zeichnen, ist es, die einzelnen Punkte direkt zu setzen. Hierzu haben wir die Kommandos PSET und PRESET. Die Position des Punktes ermittelt sich auf die gleiche Weise wie im GCURSOR-Kommando.

## Beispiele:

PSET (75,15)           setzt einen Punkt in Spalte 75, Zeile 15.

PSET (75,15),X       setzt einen Punkt, wenn die Position dunkel ist  
und löscht einen Punkt, wenn er bereits gesetzt  
war, jeweils in Spalte 75 und Zeile 15.

PRESET(75,15)       löscht den Punkt in Spalte 75, Zeile 15.

```

100 WAIT 0:DEGREE
110 FOR A=180 TO -180 STEP -2
120 B=SIN A*12                   (a)
130 C=COS A*12                   (b)
140 X=B+80                       (c)
150 Y=C+14
160 PSET(X,Y)                   (c)
170 NEXT A
180 WAIT : GPRINT               (d)

```

- (a) Parameter für einen Kreis mit dem Radius 12 und dem Mittelpunkt (0,0).
- (b) Der Mittelpunkt wird auf die Position (80,14) gesetzt.
- (c) Setzt Punkte mit den Koordinaten (X,Y).
- (d) Hält den gezeichneten Kreis.

Das Programm zeichnet einen Kreis um den Punkt (80,14) mit dem Radius 12. Der Winkel wird in Zwischenschritten von +180 bis -180 Grad mit einer FOR-NEXT-Schleife geändert. Der entsprechende Punkt wird für jeden Winkel gesetzt, so daß am Schluß ein Kreis entsteht.

## (4) Linien und Quadrate

Setzt man die Punkte mit PSET hintereinander, so erhält man eine Linie, vier Linien können ein Quadrat bilden. Wesentlich einfacher geht das aber mit dem LINE-Kommando. Zwischen zwei definierten Punkten wird vom Rechner eine Linie gezogen.

## Beispiele:

LINE (0,0)-(149,31)       zieht eine Linie zwischen den Punkten  
(0,0) und (149,31).

LINE (30,0)-(80,31),B    zeichnet ein Quadrat, dessen Diagonale  
zwischen den Punkten (30,0) und (80,31)  
liegt. Wird statt "B" "BF" gesetzt, so  
wird das Quadrat gefüllt.

LINE (30,0)-(80,31),X,BF  zeichnet ein gefülltes Quadrat mit der  
Diagonalen von (30,0) bis (80,31).  
Befinden sich in diesem Quadrat gesetzte  
Punkte, so werden sie invertiert.

```

200 "A":WAIT 0
210 LINE (60,0)-(100,31),X,BF
220 GOTO 210

```

Dieses Programm zeichnet ein gefülltes Quadrat mit der Diagonalen von (60,0) bis (100,31). Sind die Bilder, die mit den Kommandos GPRINT und PSET erzeugt wurden, noch auf dem Bildschirm, so werden sie invertiert, wenn das neue Programm mit DEF "A" aufgerufen wird.

**Achtung:** Der Bereich der graphischen Wiedergabe ist auf der linken Seite 6 Punkte breiter als der normale Wiedergabebereich für alphanumerische Zeichen.

Der PC 1350 hält die erstellten Bilder auf dem Bildschirm, bis er gelöscht wird. Am Ende eines Programms oder bei einer Unterbrechung kann es also vorkommen, daß auf der linken Seite oder im oberen Bereich Teile der Bilder stehen bleiben. Diese werden erst gelöscht, wenn die Taste **CLS** gedrückt oder das entsprechende Kommando gegeben wird. Vorher werden auch keine alphanumerischen Zeichen wiedergegeben, da der Rechner sich im Graphik-Modus befindet. Nach Drücken der Taste **CLS** ist er wieder im Normal-Modus.

#### SERIELLE E/A-FUNKTIONEN

Der PC 1350 ist mit einem seriellen E/A-Interface ausgerüstet. Hier kann der PC 1350 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 PC 1350 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 PC 1350 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.

#### DER GEBRAUCH DES SERIELLEN E/A-INTERFACE-KOMMANDOS

Sind in den PC 1350 mehrere Programme mit dem MERGE-Kommando geladen worden, so können diese nicht mit SAVE oder LOAD über das serielle E/A-Interface gesendet werden.

Das MERGE-Kommando des PC 1350 unterscheidet sich vom MERGE-Kommando anderer Computer.

**Achtung:** Das Interface des PC 1350 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 PC 1350 die Ausführung nicht selbständig unterbrechen kann. In diesem Fall drücken Sie die BRK - Taste, um den Rechner zu stoppen.

## ABKÜRZUNGEN

Der PC 1350 bietet Ihnen verschiedene Möglichkeiten, die Programmierung zu vereinfachen, indem Sie die Anzahl der Tastenbedienungen und anderer sich wiederholender Abläufe reduzieren. Eine dieser Möglichkeiten besteht darin, Befehle und Kommandos abzukürzen (siehe Kapitel 9).

In diesem Kapitel sollen vor allem zwei Möglichkeiten erörtert werden, mit deren Hilfe Sie unnötige Tastenbedienung vermeiden können: die Benutzung der **DEF**-Taste und der ReSeRve-Modus.

DIE **DEF**-TASTE UND DER RESERVE-MODUS

Es wird häufiger vorkommen, daß Sie mehrere verschiedene Programme zur gleichen Zeit im Speicher Ihres PC 1350 abspeichern wollen. (Vergessen Sie nicht, verschiedene Zeilennummern zu vergeben!) Um ein Programm mit dem RUN- oder GOTO-Befehl zu starten, müssen Sie sich normalerweise an die erste Zeilennummer eines jeden Programmes erinnern (siehe Kapitel 9). Aber es gibt eine einfachere Möglichkeit! Sie können jedes Programm mit einem einzelnen Buchstaben benennen und das Programm dann mit nur zwei Tastenbedienungen starten.

Anmerkung: Setzen Sie den Kennbuchstaben, auf den Sie sich später beziehen wollen, in die erste Programmzeile. Der Kennbuchstabe muß in Anführungszeichen eingeschlossen und von einem Doppelpunkt gefolgt sein.

```
10: "A":PRINT "ERSTES"
20: END
80: "B":PRINT "ZWEITES"
90: END
```

Jedes der folgenden Zeichen kann als Kennbuchstabe benutzt werden: A, S, D, F, G, H, J, K, L, =, Z, X, C, V, B, N, M und die Leertaste (SPC). Wie Sie sehen, sind dies alle Tasten in den unteren beiden Zeilen der Tastatur. Dieser Teil wurde zur Unterscheidung gegenüber dem Rest des Tastenfeldes durch eine Linie getrennt.

Anmerkung: Um das Programm ablaufen zu lassen, brauchen Sie nun nicht RUN 80 oder GOTO 10 einzugeben. Sie drücken nur die **DEF**-Taste und dann den Kennbuchstaben. Im obigen Beispiel würde, wenn Sie **DEF** und dann "B" drücken würden, "ZWEITES" in der Anzeige erscheinen.

Wenn Sie **DEF** benutzen, um ein Programm zu starten, werden Variable und Modus-Einstellungen ebenso verarbeitet, als wenn Sie GOTO benutzen würden. In Kapitel 9 finden Sie weitere Einzelheiten.

Um ein anderes Programm mit der **DEF**-Taste während der Ausführung eines Input-Befehls zu starten, drücken Sie die **ENTER**- oder **BRK**-Taste, um den Input-Befehl aufzuheben und das neue Programm zu starten.



## RESERVE-MODUS

Ein anderer Weg, mit dem PC 1350 Zeit einzusparen, ist der RESERVE-Modus.

Innerhalb des Speichers Ihres PC 1350 sind 144 Zeichen für den "Reserve-Speicher" vorgesehen. Sie können diesen Speicherteil benutzen, um häufig gebrauchte Ausdrücke zu speichern. Um diese wieder abzurufen, brauchen Sie nur zwei Tasten zu bedienen.

## Anmerkung:

- \* Sie speichern Ausdrücke im RESERVE-Modus ab, der Wiederabruf zum Gebrauch erfolgt aber im RUN- oder PROgramm-Modus.
- \* Der MEM-Befehl kann abgespeichert werden, er kann aber nicht ausgeführt werden.
- \* Der PC 1350 hat einen Reserve-Speicher von 144 Bytes. Sie können also bis zu 144 Bytes in den Reserve-Speicher eingeben. Ein BASIC-Befehl, eine Funktion, ein Buchstabe oder eine Zahl sind jeweils ein Byte.

Beispiel: A: S: + - 1 2 A B SIN COS INPUT RUN ...  
(je 1 Byte)

- \* Die Länge der Strings, die pro Taste abgespeichert werden können, beträgt maximal 80 Tastenbedienungen - einschließlich der **ENTER**-Taste.

Versuchen Sie, das folgende Beispiel nachzuvollziehen:

Geben Sie **SHIFT** und **MODE** ein. Beachten Sie, daß die Modus-Anzeige "RUN" und "PRO" verschwindet und die Nachricht "RESERVE MODE" auf dem Bildschirm erscheint, wenn Sie den RESERVE-Modus eingeschaltet haben.

Geben Sie NEW ein und drücken dann die **ENTER**-Taste. Dadurch löschen Sie alle vorher gespeicherten Daten, so, wie Sie mit NEW im PROgramm-Modus alle vorher gespeicherten Programme löschen.

Geben Sie **SHIFT** gefolgt von "A" ein:

EINGABE

ANZEIGE

**SHIFT** A

A: \_

Beachten Sie, daß das "A" links in der Anzeige erscheint und von einem Doppelpunkt gefolgt wird.

Geben Sie das Wort "PRINT" ein und betätigen Sie die **ENTER**-Taste.

EINGABEANZEIGEPRINT 

A: PRINT

Ein Leerschritt erscheint nach dem Doppelpunkt und zeigt an, daß "PRINT" nun im Reservespeicher unter dem Buchstaben "A" gespeichert worden ist.

Schalten Sie den PC 1350 in den PROgramm-Modus um. Geben Sie NEW, gefolgt von ENTER ein, um den Programmspeicher zu löschen. Geben Sie 10 als Zeilennummer ein und betätigen Sie dann die  -Taste und die "A"-Taste:

EINGABEANZEIGE10  A

10 PRINT \_

10:PRINT

Sofort wird das Wort "PRINT" hinter der Zeilennummer in der Anzeige erscheinen.

Im Reserve-Speicher kann jede beliebige Zeichenfolge gespeichert werden. Die gespeicherten Strings können jederzeit im RUN- oder PROgramm-Modus wieder abgerufen werden, indem Sie  drücken und anschließend den Buchstaben, unter dem Sie den String abgespeichert haben. Die dafür zur Verfügung stehenden Tasten sind dieselben wie im Zusammenhang mit  erwähnt, also alle Tasten im dunkleren Bereich des Tastenfeldes.

Um eine gespeicherte Zeichensequenz zu editieren, schalten Sie den Reserve-Modus ein und drücken Sie  , gefolgt von dem Buchstaben, unter dem die Sequenz gespeichert ist. Mit Hilfe der Pfeile (rechts und links), der  - und der  -Taste können Sie nun genauso editieren wie in den anderen Modi.

Ist das letzte Zeichen in einer abgespeicherten Sequenz das "@"-Zeichen, wird es bei Wiederabruf der Sequenz als  ausgegeben. Speichern sie beispielsweise den String "GOTO 100@" unter der "G"-Taste, so wird bei Betätigung von  und "G" im RUN-Modus augenblicklich die Ausführung des Programms in Zeile 100 beginnen. Ohne das "@"-Zeichen müssen Sie nach der  "G"-Sequenz  eingeben, um das Programm zu starten.

Um Reservespeicher zu löschen:

1. Die Tastenfolge NEW und  löscht alle Speicher. Denken Sie bitte daran, daß die oben beschriebene Operation im RESERVE-Modus durchgeführt werden muß.
2. Um einen Reserve-Speicher zu löschen, benutzen Sie die  oder  -Taste, wie unten beschrieben:

Beispiel: Löschen Sie A\*A, was unter der Taste S gespeichert ist.

Eingabe	Anzeige	Bemerkungen
SHIFT S	S: _	Reserve-Modus
A * A ENTER	S: A * A	
CLS	>	
SHIFT S	S: A * A	
▶ oder ◀	S: A * A	
DEL DEL DEL	S: _	Löscht A * A.
ENTER	>	

### SCHABLONEN

Zu Ihrem PC 1350 gehören zwei Schablonen. Sie können diese benutzen, um das Erinnern an häufig gebrauchte Reserve-Sequenzen oder Kennbuchstaben-Zuweisungen über DEF zu erleichtern. Nachdem Sie den Programmen Kennbuchstaben zugewiesen haben oder Sequenzen eingerichtet haben, markieren Sie die Schablonen, damit Sie wissen, welche Kennung zu welcher Taste gehört. Sie können dann mit nur zwei Tastenbetätigung Programme abarbeiten lassen oder Sequenzen abrufen.

Wenn Sie beispielsweise eine Reihe von Programmen haben, die Sie häufig zur gleichen Zeit benutzen, vergeben Sie Kennbuchstaben und markieren die Schablonen, so daß Sie die Programmausführung ganz einfach mit nur zwei Tastenbedienungen starten können. Sie können auch häufig gebrauchte BASIC-Befehle und Kommandos in den Reserve-Speicher einspeisen und eine Schablone entsprechend markieren, um so die Eingabe von BASIC-Programmen zu beschleunigen.

## KAPITEL 7

## DER GEBRAUCH DES CE-126P DRUCKER/CASSETTENRECORDER-INTERFACE

Das Zusatzgerät CE-126P Drucker/Cassettenrecorder-Interface erlaubt den Anschluß eines Druckers oder Cassettenrecorders an Ihren PC 1350.

Die Charakteristika des CE-126P sind:

- \* Thermodrucker mit 24 Zeichen Schreibbreite
- \* Handliche Papierzuführungs- und Abreißvorrichtung
- \* Gleichzeitiges Ausdrucken von Rechnungen, wenn gewünscht
- \* Einfache Kontrolle der Anzeige oder Druckerausgabe in BASIC
- \* Eingebautes Cassetten-Interface mit REMOTE-Funktion
- \* Manuelle oder programmgesteuerte Kontrolle des Recorders zum Abspeichern von Programmen und Daten
- \* Stromversorgung über Trockenbatterien für tatsächliche Tragbarkeit

Um den PC 1350 an das CE-126P anzuschließen, bedienen Sie sich bitte des mit dem CE-126P gelieferten Handbuches.

## DER GEBRAUCH DES DRUCKERS

Wenn Sie mit dem PC 1350 Rechnungen ohne Programmunterstützung durchführen, können Sie diese über den CE-126P gleichzeitig ausdrucken lassen.

Sie erreichen dies durch einfaches Drücken der **SHIFT**-Taste und anschließend der **ENTER**-Taste, wobei sich der Computer im RUN-Modus befinden muß.

Wenn Sie nun am Ende einer Berechnung die **ENTER**-Taste drücken, wird der Inhalt der Anzeige vom Drucker ausgegeben. Dabei steht die Berechnung selbst in der ersten Zeile, das Ergebnis erscheint in der zweiten. Zum Beispiel:

EINGABE

300/50 **ENTER**

AUSDRUCK

300/50

6.

Mit dem LPRINT-Befehl können Sie auch Teile von BASIC-Programmen ausdrucken lassen (Einzelheiten dazu finden Sie in Kapitel 8). LPRINT kann in derselben Weise wie der PRINT-Befehl benutzt werden. Der Unterschied besteht darin, daß Sie, wenn Sie einen Ausdruck von mehr als 24 Zeichen mit PRINT drucken wollen, die über 26 hinausgehenden Zeichen nicht sehen können. Mit dem LPRINT-Befehl werden diese Zeichen in einer zweiten und, wenn erforderlich, einer dritten Zeile ausgegeben.

Programme, die mit einem PRINT-Befehl geschrieben wurden, können modifiziert werden, indem Sie in das Programm ein PRINT=LPRINT-Statement hineinschreiben (Einzelheiten siehe Kapitel 9). Alle diesem Statement folgenden PRINT-Befehle werden nun wie LPRINT-Befehle gewertet. Durch PRINT=PRINT heben Sie das Statement wieder auf. Diese Struktur kann auch mit einem IF-Statement in ein Programm geschrieben werden, wodurch Sie sich zum Zeitpunkt der Programmdurchführung eine Auswahlmöglichkeit schaffen.

Sie können Ihr Programm mit dem LLIST-Kommando auflisten lassen (Einzelheiten siehe Kapitel 9). Wenn Sie LLIST ohne Zeilennummern verwenden, werden alle in dem Moment im Speicher befindlichen Programme nach der numerischen Reihenfolge ihrer Zeilennummern ausgegeben. Die auszugebenden Zeilen lassen sich begrenzen, indem Sie die entsprechende Spanne mit dem LLIST-Kommando eingeben. Wenn Programmzeilen länger als 24 Zeichen sind, werden für den Ausdruck zwei oder mehr Zeilen benutzt. Die zweite und alle folgenden Zeilen werden um vier oder sechs Zeichen eingerückt, so daß die jeweilige Zeilennummer eindeutig zu identifizieren ist. (Von Zeile 1 bis 999: Einrückung um 4 Zeichen, ab Zeile 999 Einrückung um 6 Zeichen.)

#### Achtung:

- \* Falls eine Fehlermeldung ERROR 8 infolge eines Papierstaus auftritt, reißen Sie das Papier ab und ziehen Sie das verbliebene Papier vollständig aus dem Drucker. Drücken Sie dann die **[CLS]** -Taste, um die Fehlermeldung zu löschen.
- \* Wenn der Drucker starken externen elektrischen Störfeldern ausgesetzt ist, können eventuell falsche Zahlen willkürlich ausgedruckt werden. In diesem Fall drücken Sie die **[BRK]** -Taste, um den Druckvorgang zu unterbrechen. Schalten Sie die Stromversorgung des CE-126P ab und wieder an, drücken Sie dann die **[CLS]** -Taste. Durch Druck auf die **[CLS]** -Taste initialisieren Sie den Drucker. Wenn der Drucker einen Papierstau verursacht oder er während des Druckens starken elektrischen Schwankungen ausgesetzt ist, kann es passieren, daß der Drucker nicht normal arbeitet und lediglich "BUSY" in der Anzeige erscheint. Wenn das passiert, schalten Sie den Drucker ab, indem Sie die **[BRK]** -Taste drücken. Beseitigen Sie den Papierstau. Schalten Sie den CE-126P ab und dann wieder an und drücken Sie die **[CLS]** -Taste.
- \* Wenn der CE-126P nicht in Betrieb ist, schalten Sie bitte den Drucker ab, um die Batterien zu schonen.

#### DER GEBRAUCH DES CASSETTEN-INTERFACE

Mit Hilfe dieses Cassetten-Interface können Sie Daten und Programme aus dem Computer auf Tonbandcassetten abspeichern (natürlich brauchen Sie dazu auch einen Cassettenrecorder wie das von uns zu diesem Taschencomputer-System gelieferte Modell CE-152). Sind die Daten und Programme einmal auf Band gespeichert, können Sie sie durch einen einfachen Ablauf wieder in den Computer einladen.

## DAS ANSCHLIESSEN DES CE-126P AN EINEN CASSETTENRECORDER

Hier sind nur drei Verbindungen zu schaffen:

1. Stecken Sie den roten Klinkenstecker in die MICrofonbuchse des Cassettenrecorders.
2. Stecken Sie den grauen Klinkenstecker in die Kopfhörerbuchse am Cassettenrecorder.
3. Stecken Sie den schwarzen Stecker in die REMote-Buchse am Cassettenrecorder.

## CASSETTENRECORDER

Wir empfehlen Ihnen, für Ihr Taschencomputer-System das Zusatzgerät CE-152 zu benutzen. Dieses Gerät ist auf den PC 1350 zugeschnitten und nimmt Daten und Programme über das CE-126P Interface auf. Jedes abgespeicherte Programm kann wieder aufgesucht und in den PC 1350 eingeladen werden.

Wenn Sie einen anderen Recorder benutzen als die Option CE-152, muß dieser den folgenden Mindestanforderungen entsprechen, um über das Interface CE-126P angeschlossen werden zu können:

## Anforderungen

- |                              |  |
|------------------------------|--|
| 1. Recordertyp               | Jedes Cassettengerät, ob mit Standard- oder Microcassetten, kann benutzt werden, sofern die unten aufgeführten Bedingungen erfüllt sind. |
| 2. Aufnahmeanschluß          | Der Recorder sollte über einen Klinkensteckeranschluß (2,5 mm) mit der Bezeichnung MIC verfügen. Benutzen Sie niemals den AUX-Anschluß.  |
| 3. Aufnahmeimpedanz          | Der Aufnahmeanschluß sollte eine niedrige Impedanz besitzen (200 - 1.000 Ohm).   |
| 4. Minimales Aufnahme-niveau | Unter 3mV oder -50dB.  |
| 5. Wiedergabeanschluß        | Sollte ein Klinkensteckeranschluß (2,5 mm) mit der Bezeichnung "EXT" (EXTerner Lautsprecher), "MONITOR", "EAR" (Kopfhörer) sein.         |
| 6. Wiedergabeimpedanz        | Sollte unter 10 Ohm liegen.  |
| 7. Wiedergabeniveau          | Sollte über 1V liegen (maximale Wiedergabe über 100 mW).   |
| 8. Verzerrung                | Sollte innerhalb 15 %, innerhalb eines Bereiches zwischen 2 und 4 KHZ liegen.  |

### 9. Gleichlaufschwankung maximal 0.3 % (W.R.M.S.)

- \* Falls der mit dem CE-126P Interface gelieferte Klinkestecker nicht zu dem Eingang an Ihrem Gerät paßt, können Sie im Fachhandel entsprechende Adapter bekommen.

Anmerkung: Einige Cassettengeräte können aufgrund spezieller Abweichungen von den Spezifikationen nicht anzuschließen sein. Weiterhin können mit Geräten, die infolge langjährigen Gebrauchs ihre elektrischen Eigenschaften verändert haben (starke Verzerrung, erhöhtes Rauschen, Spannungsabfall), unbefriedigende Ergebnisse erzielt werden.

### DAS ARBEITEN MIT CASSETTEN-INTERFACE UND RECORDER ABSPEICHERN AUF MAGNETBAND

(Siehe auch Anmerkungen zum Tonband.)

1. Schalten Sie den REMOTE-Schalter des CE-126P auf OFF.
2. Geben Sie ein Programm oder Daten in den Computer ein.
3. Legen Sie die Cassette in den Recorder ein. Bestimmen Sie die Stelle auf dem Band, an der Sie das Programm abspeichern wollen.
  - \* Wenn Sie ein leeres Band benutzen, achten Sie darauf, daß Sie den Leervorspann aus nicht-magnetisierter Folie nicht bespielen.
  - \* Wenn Sie ein bereits teilweise bespieltes Band benutzen, suchen Sie eine Stelle, an der noch keine Daten gespeichert sind!
4. Stecken Sie den roten Stecker des Interface in die MICrofonbuchse des Cassettenrecorders und den schwarzen Stecker in die REM-Buchse.
5. Bringen Sie den REMOTE-Schalter in die Stellung ON.
6. Zur Aufnahme drücken Sie nun am Cassettenrecorder die Aufnahme- und die PLAY-Taste gleichzeitig.
7. Geben Sie die Aufnahmeinstruktionen ein (CSAVE-Kommando, PRINT#-Kommando und drücken Sie die ENTER -Taste).

Setzen Sie die Einheit zunächst in den RUN- oder PRO-Modus und drücken Sie die folgenden Tasten:

C S A V E SHIFT " dateiname SHIFT " ENTER

(Um den Inhalt des Datenspeichers auf Band zu überspielen, drücken Sie die folgenden Tasten:

P R I N T    SHIFT # ENTER )

z. B.:

C S A V E    SHIFT " AA SHIFT "    ENTER

Wenn Sie die **ENTER** -Taste drücken, wird das Band zu laufen beginnen, wobei ein etwa 8 Sekunden langes Stück unbespielt bleibt. (Ein Piepton ist zu hören.) Anschließend werden der Dateiname und sein Inhalt aufgenommen.

- 8) Wenn die Aufnahme vollständig ist, erscheint das Bereitschaftssymbol in der Anzeige (>) und der Recorder hält automatisch an. Nun haben Sie Ihr Programm im Cassettenrecorder abgespeichert. Es befindet sich jedoch trotzdem im Computer.

Wenn Daten über einen Programmbefehl automatisch abgespeichert werden sollen (PRINT#-Befehl, nicht bei manueller Bedienung), befolgen Sie erst die Schritte 1 bis 6, ehe Sie das Programm ablaufen lassen.

Als Hilfe beim Auffinden bestimmter Programme auf Band benutzen Sie bitte das Bandzählwerk des Recorders.

#### VERGLEICHEN VON SPEICHER- UND BANDINHALTEN

(Siehe auch Anmerkungen zum Tonband.)

Nachdem Sie ein Programm von Band geladen bzw. auf Band abgespeichert haben, können Sie überprüfen, ob das Programm im Computer identisch mit dem auf Band ist (und so sicher gehen, daß alles in Ordnung ist, ehe Sie mit dem Programmieren oder der Programmdurchführung fortfahren).

1. Schalten Sie den REMOTE-Schalter auf OFF.
2. Bewegen Sie das Band mit den Vor- und Rücklauftasten an die Stelle kurz vor Beginn des zu prüfenden Programms.
3. Stecken Sie den grauen Stecker in die Kopfhörerbuchse und den schwarzen Stecker in die REMote-Buchse.
4. Schalten Sie den REMOTE-Schalter auf ON.
5. Drücken Sie die PLAY-Taste des Recorders.
6. Geben Sie ein CLOAD?-Kommando ein und beginnen Sie den Prüfungsvorgang mit der **ENTER** -Taste. Gehen Sie dabei folgendermaßen vor:

Schalten Sie den RUN- oder PRO-Modus ein.



Geben Sie folgende Tastenfolge ein:

**C** **L** **O** **A** **D** **SHIFT** **?** **SHIFT** **"** **AA** **SHIFT** **"** **ENTER**

↙ Der Dateiname, den Sie vorher gewählt haben.

Der Taschencomputer wird nun automatisch nach dem vorgegebenen Dateinamen suchen und wird den Inhalt des Tonbandes mit dem Speicherinhalt vergleichen.

Während des Vergleichsvorganges erscheint das Zeichen "\*" an der äußersten rechten Stelle der unteren Display-Zeile. Das Zeichen "\*" erlischt, wenn der Vergleich beendet ist. Solange der Computer nach dem Dateinamen sucht, wird das "\*" -Zeichen nicht in der Anzeige ausgegeben, da der eigentliche Vergleich noch nicht begonnen hat. (Das gleiche geschieht, wenn das erste Programm ohne Dateinamen gelesen wird.)

Werden die Programme als übereinstimmend erkannt, erscheint das Bereitschaftssymbol (>) in der Anzeige des Taschencomputers.

Decken sich die Programme nicht, wird der Vergleich unterbrochen und die Meldung ERROR 8 ausgegeben. Wenn dies geschieht, versuchen Sie es bitte noch einmal.

#### EINLADEN VON MAGNETBAND

(Siehe auch Anmerkungen zum Tonband.)

Um Programme oder Daten vom Magnetband in den Computer einzuladen, gehen Sie bitte folgendermaßen vor:

1. Schalten Sie den REMOTE-Schalter auf OFF.
2. Legen Sie das Band in den Recorder ein. Bringen Sie das Magnetband mit Hilfe der Vor- bzw. Rückkluftasten kurz vor die Stelle, an der das einzulesende Programm beginnt.
3. Stecken Sie den grauen Stecker in die Kopfhörerbuchse des Recorders und den schwarzen Stecker in die REM-Buchse. (Falls Sie einen Recorder benutzen, der keinen REM-Eingang hat, drücken Sie für eine kurze Unterbrechung die PAUSE-Taste.)
4. Schalten Sie den REMOTE-Schalter wieder auf ON.
5. Drücken Sie den PLAY-Schalter des Recorders, um die Einheit in den Playback-Modus zu versetzen. Stellen Sie VOLUME auf mittlere oder maximale Lautstärke ein. Drehen Sie die Höhen maximal auf.
6. Geben Sie die Übertragungsanweisungen (CLOAD-Kommando, INPUT#-Kommando) ein und starten Sie die Ausführung mit **ENTER**.

Setzen Sie die Einheit in den RUN-Modus, danach betätigen Sie die folgenden Tasten:

**C** **L** **O** **A** **D** **SHIFT** **"** dateiname **SHIFT** **"** **ENTER**

(Um den Inhalt den Datenspeichers zu laden, drücken Sie:

**I** **N** **P** **U** **T** **SHIFT** **#** **ENTER** )

Z. B.:

**C** **L** **O** **A** **D** **SHIFT** **"** AA **SHIFT** **"** **ENTER**

Der vorgegebene Dateiname wird nun automatisch gesucht und sein Inhalt wird in den Taschencomputer eingeladen. Das Zeichen "\*" erscheint, solange der Ladevorgang abläuft. Das gleiche geschieht, wenn das erste Programm ohne Dateinamen gelesen wird.

Das Zeichen "\*" erlischt, wenn das Programm vollständig geladen ist.

7. Wenn das Programm übertragen worden ist, hält der Computer automatisch den Bandlauf an und das Bereitschaftssymbol (>) erscheint in der Anzeige.

Um im Verlauf eines Programms Daten zu übertragen (INPUT#-Befehl), führen Sie erst die Schritte 1 bis 5 aus, ehe Sie das Programm starten.

#### Anmerkungen:

- \* Wenn Sie eine Fehlermeldung erhalten (ERROR 8), fangen Sie noch einmal von vorn an. Wiederholt sich die Fehlermeldung, verändern Sie vorsichtig die VOLUME-Einstellung.
- \* Wird keine Fehlermeldung ausgegeben und läuft das Band weiter (während der Computer die Anzeige BUSY ausgibt), verläuft die Übertragung fehlerhaft. Drücken Sie dann die **BK**-Taste, um die Übertragung abzurechnen und wiederholen Sie die Schritte.
- \* Bleibt die Fehlermeldung bestehen oder läuft das Band auch nach mehreren Korrekturversuchen weiter, reinigen und entmagnetisieren Sie den Tonkopf des Recorders.

#### ANMERKUNGEN ZUM TONBAND

1. Benutzen Sie zur Datenübertragung oder zum Vergleich ausschließlich den Recorder, mit dem die Programme aufgezeichnet wurden. Die Verwendung eines anderen Gerätes kann möglicherweise die Datenübertragung oder den Programmvergleich ausschließen.
2. Benutzen Sie grundsätzlich Tonbandcassetten der höchsten Qualität, um Daten und Programme darauf abzuspeichern (Bandmaterial zum "Sparpreis" weist u. U. nicht die für digitale Aufzeichnungen erforderlichen Eigenschaften auf!).

3. Halten Sie die Tonköpfe und das Cassettenlaufwerk sauber - benutzen Sie eine Reinigungscassette zur Reinigung.
4. Lautstärke-Einstellung: mittlere oder maximale Lautstärke. Die Lautstärke-Einstellung kann von großer Bedeutung beim Lesen von Daten vom Band sein. Ändern Sie die Einstellung geringfügig, wenn dies erforderlich wird. Schon geringe Änderungen können zu hervorragenden Ergebnissen führen.
5. Achten Sie darauf, daß alle Verbindungen zwischen Computer und Cassetten-Interface sicher sind. Dasselbe gilt für die Verbindungen zwischen Interface und Recorder. Darüber hinaus müssen die Verbindungselemente sauber und staubfrei sein.
6. Sollte es beim Gebrauch des AC-Netzteils Probleme geben, speisen Sie den CE-126P und/oder den Recorder über Batterien (manchmal kommt durch die Benutzung des AC-Netzteils ein Summen zu den zu übertragenden Signalen, das die eigentliche Aufnahme stört.)
- \* Wenn Sie das AC-Netzteil an den CE-126P anschließen, schalten Sie die Stromversorgung des CE-126P ab und stellen Sie die Verbindung erst dann her.
7. Höhenregler: auf maximale Höhen einstellen.
8. Wenn Sie Programme oder Daten auf einem bereits gebrauchten Band abspeichern wollen, löschen Sie diesen Teil vor der Eingabe und geben Sie dann den Speicherbefehl. (Vergewissern Sie sich, daß das alte Programm restlos gelöscht ist.)

## KAPITEL 8

## DIE RAM-KARTE (optional)

Der Speicher des PC 1350 kann mit den RAM-Karten CE-201M oder CE-202M erweitert werden. Der Programm-/Datenspeicher wird damit um 8 bzw. 16 KByte erweitert.

Selbst wenn die RAM-Karte aus dem PC 1350 entfernt wird, bleibt ein in ihr gespeichertes Programm erhalten, so daß man also Programme auf verschiedenen RAM-Karten bereitliegen haben kann und das gewünschte Programm durch Wechseln der RAM-Karte in den Rechner einlädt.

## 1. Installation der RAM-Karte

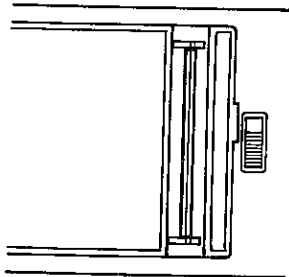
Wird die RAM-Karte das erste Mal eingesetzt, sollten Sie darauf achten, daß die Batterie eingesetzt ist.

- (1) Schalten Sie Ihren PC 1350 aus.

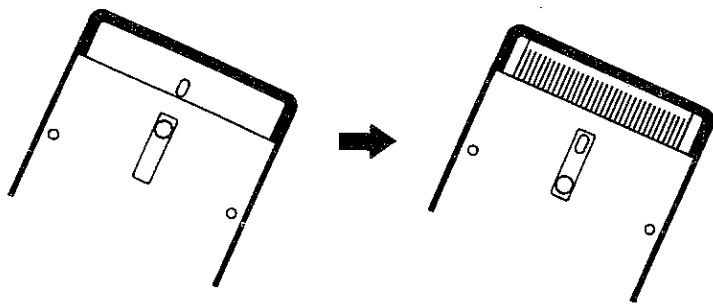
Der PC 1350 muß immer abgeschaltet sein, wenn die RAM-Karte eingesetzt oder herausgenommen werden soll.

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

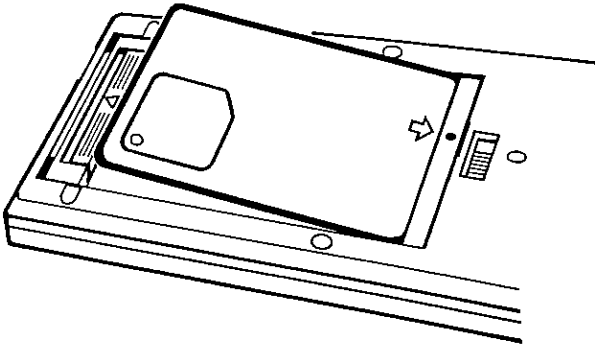
Achten Sie bitte darauf, keine Bauteile oder Kontakte im Inneren des PC 1350 zu berühren.



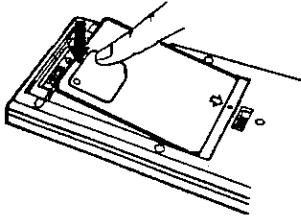
- (3) Öffnen Sie die Kontaktschutzhaube der RAM-Karte. Achten Sie bitte darauf, daß die Haube ganz geöffnet ist und berühren Sie keinen der Kontakte.



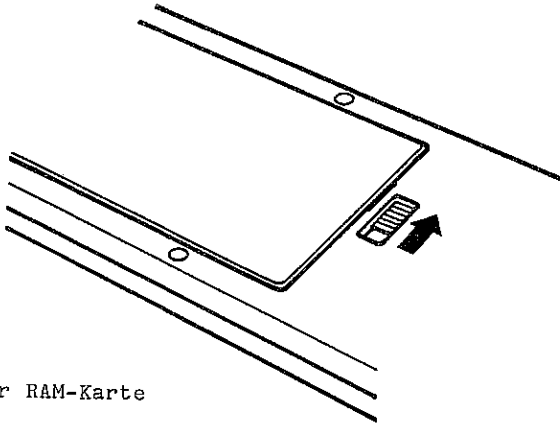
- (4) Schieben Sie die Karte mit der Kontaktseite voran in den PC 1350, wie in der Zeichnung dargestellt.



Schieben Sie die Karte nie anders herum in den Rechner und nie mit geschlossener oder nur teilweise geöffneter Schutzhaube. Sowohl die Karte als auch der PC 1350 könnten zerstört werden. Drücken Sie dann auf das Gehäuse der Karte, bis sie einrastet.

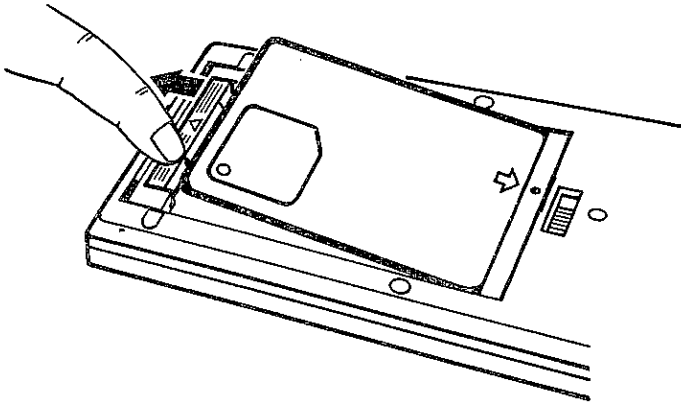


- (5) Setzen Sie die Klappe des PC 1350 wieder ein und arretieren Sie sie, indem Sie den Hebel wieder in die Position "LOCK" schieben. Der PC 1350 wird nicht arbeiten, wenn die Arretierung nicht ganz in die "LOCK"-Position geschoben wurde, selbst wenn alle Batterien korrekt eingesetzt sind. Sollte das Gerät nicht einwandfrei funktionieren, schalten Sie den PC 1350 noch einmal ab, überprüfen die Arretierung der Klappe auf der Rückseite und schalten ihn dann wieder ein.



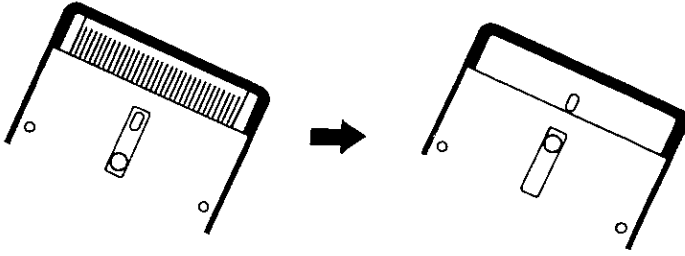
## 2. Entnahme der RAM-Karte

- (1) Schalten Sie Ihren PC 1350 aus.
- (2) Entfernen Sie die Klappe auf der Rückseite des Rechners wie oben beschrieben (1.(2)). Schieben Sie dann die Kartenarretierung in Pfeilrichtung, um die RAM-Karte zu lösen.

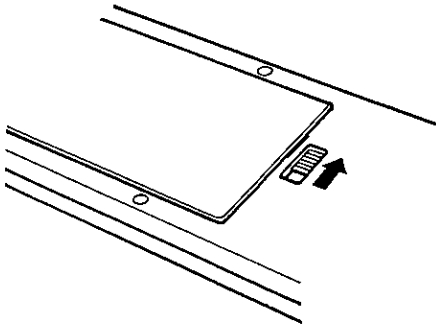


- (3) Schließen Sie bitte sofort die Kontaktschutzhaube der RAM-Karte, nachdem sie herausgenommen ist.

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



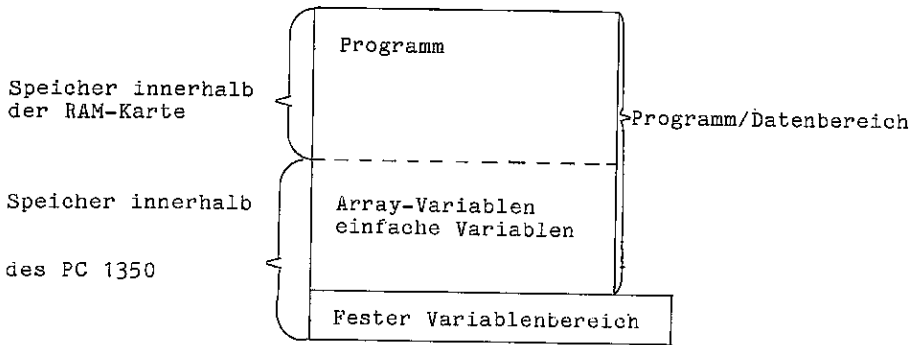
- (4) Setzen Sie die Klappe des PC 1350 wieder ein und schieben Sie die Arretierung wieder in die "LOCK"-Position.



### 3. Einsatz der RAM-Karte

Ist der Speicher der RAM-Karte frei, so steht Ihnen die Summe der Speicher des Rechners und der Karte für Programme und Daten zur Verfügung.

Wie Sie aus der Illustration ersehen können, werden Programme in der RAM-Karte abgelegt. Array- und einfache Variable aber in dem Speicher des PC 1350.



Das Programm in der RAM-Karte bleibt auch dann erhalten, wenn die Karte aus dem PC 1350 entnommen wird, da die Karte mit eigenen Batterien ausgerüstet ist. Deshalb kann das Programm zusammen mit der Karte erneut in den PC 1350 "eingeladen" werden, indem Sie einfach die Karte wieder in den Rechner einstecken.

**Achtung:** Überschreitet ein Programm die Kapazität der RAM-Karte, wird das Programm bei Entnahme der Karte zerstört und kann mit der Karte nicht wieder in den PC 1350 eingeladen werden!

Achten Sie darauf, daß ein Programm die Kapazität der Karte nicht überschreitet, wenn Sie es in der Karte auf Dauer speichern wollen.

**Achtung:** Der Reservespeicherbereich des PC 1350 kann nicht in die RAM-Karte umgeladen werden.

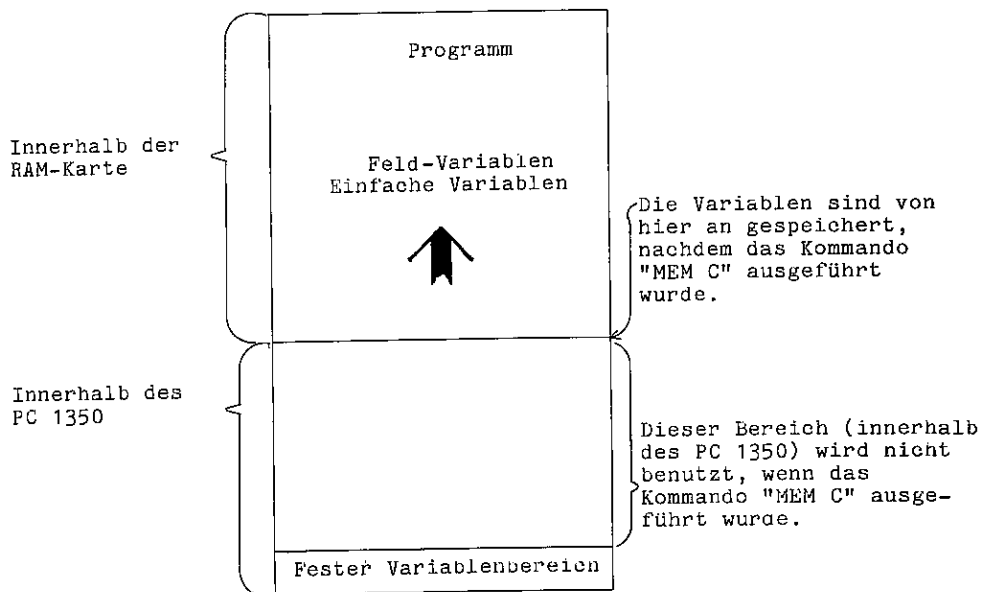
Kapazitäten des Programm-/Datenspeichers

PC 1350 + CE-201M	CE-201M allein	PC 1350 + CE-202M	CE-202 allein
11.262 Byte	8.142 Byte	19.454 Byte	16.334 Byte

4. Datensicherung in der RAM-Karte

Wie Sie in einer der vorigen Illustrationen erkennen konnten, werden Variable, wie z. B. einfache Variable oder Felder, im Speicher des PC 1350 abgelegt. Wird nun die RAM-Karte aus dem PC 1350 entfernt, so werden die Variablen natürlich gelöscht. Mit dem Kommando "MEM C" können die Variablen in die RAM-Karte übertragen werden. Die Variablen, auch Daten genannt, bleiben auch dann erhalten, wenn die RAM-Karte aus dem PC 1350 herausgenommen wird.





\* Gelöscht wird das "MEM C"-Kommando mit dem Kommando "MEM B". Außerdem muß man berücksichtigen, daß die Variablen im Speicher des PC 1350 gelöscht werden, wenn das "MEM C"-Kommando angewandt wird.

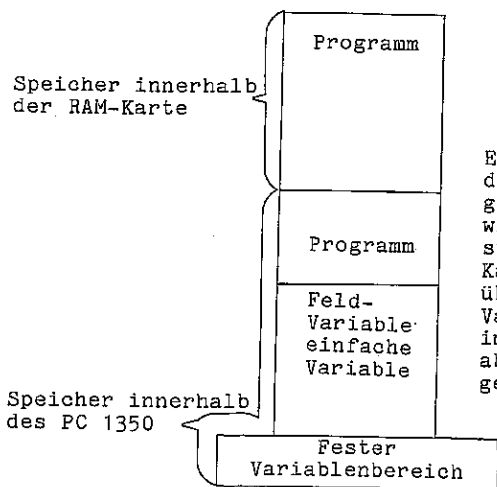
**Achtung:** Überschreitet das Programm die Kapazität der RAM-Karte, wird das Kommando "MEM C" automatisch gelöscht und durch "MEM B" ersetzt.

Welches der beiden Kommandos "MEM C" oder "MEM B" gesetzt ist, wird in der RAM-Karte gespeichert. Bei Entnahme aus dem Rechner wird das Kommando für den PC 1350 unwirksam. Re-Installation setzt das Kommando wieder in Kraft.

Um die Daten aus der RAM-Karte zurück zu erhalten, muß eine Taste mit dem Programm belegt werden. Rufen Sie dann das Programm mit dieser Taste auf. Wird das Programm mit dem Kommando "RUN" gestartet, werden die Variablen gelöscht. Benutzen Sie deshalb für ein solches Programm nie das "RUN"-Kommando. Konstante, oft auch feste Variable genannt, können nicht in der RAM-Karte gespeichert werden. Wenn Sie also Daten sichern wollen, müssen Sie einfache oder Feld-Variable benutzen.

## 5. Vorbedingungen für die RAM-Karte

- \* Wenn ein Programm im Speicher des PC 1350 gespeichert ist und Sie dann eine RAM-Karte installieren, so wird das Programm im Speicher des PC 1350 ignoriert. Es kann weder aufgerufen noch gestartet werden. Ein Programm, das die Kapazität der Karte überschreitet, zerstört ein im Speicher des Rechners enthaltenes Programm. Auch wenn das Programm der Karte seine Variablen im Speicher des Rechners ablegt, wird ein Programm im PC 1350 gelöscht.



Ein Programm, das vor Einsetzen der RAM-Karte im PC 1350 gespeichert wurde. Dieses Programm wird ignoriert. Es wird aber zerstört, wenn das Programm aus der Karte die Kapazität der Karte überschreitet. Auch wenn die Variablen des Programms der Karte im Speicher des PC 1350 abgelegt werden, wird das dort gespeicherte Programm zerstört.

Ist das Programm im PC 1350 nicht zerstört worden, kann es natürlich nach Entfernen der RAM-Karte wieder gestartet werden.

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

Wird die Batterie der RAM-Karte aber gewechselt, während die Karte in den PC 1350 eingesetzt ist, so bleiben alle Daten und Programme der Karte erhalten.

Die Daten und Programme werden von einer Batterieladung für etwa 34 Monate bei der CE-201M und etwa 18 Monate bei der CE-202 gehalten. Die Zeiten, in denen die Karte im PC 1350 eingesetzt ist, zählen dabei nicht mit.

Achtung: Drücken Sie nie die **RESET** -Taste an der Rückseite des PC 1350, wenn eine Karte eingesetzt wurde. Die **RESET** -Taste löscht dann alle Daten und Programme, die in der RAM-Karte gespeichert sind. Sollte es trotzdem nötig sein, die **RESET** -Taste zu drücken, so lesen Sie bitte die entsprechenden Kapitel im Handbuch, wie die Daten der Karte erhalten bleiben.

## KAPITEL 9

### BASIC REFERENZ TEIL

Das folgende Kapitel ist in drei Teile untergliedert:

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

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

Graphische Darstellungen	Seite 9- 82
Serieller E/A-Anschluß	Seite 9- 93
Textfunktionen	Seite 9-107

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 den 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

CONT            NEW  
GOTO\*           RUN

##### Variablenkontrolle

CLEAR\*  
DIM\*  
MEM\*

##### Cassettenkontrolle

CLOAD            INPUT# \*  
CLOAD?           MERGE  
CSAVE            PRINT# \*

##### Winkelmodus

DEGREE\*            RADIAN\*  
GRAD                #

## Debug-Kontrolle

LIST           TROFF\*  
LLIST          TRON\*

## Andere

BEEP\*            USING\*  
PASS             WAIT\*  
RANDOM\*

## Graphik

GPRINT\*         PRESET\*  
LINE\*          PSET\*

## Serieller E/A-Anschluß

CLOSE\*           LOAD\*  
CONSOLE\*        OPEN\*  
INPUT#1\*        PRINT#1\*  
LLIST            SAVE

## Text

BASIC  
TEXT

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

BEFEHLE

## Kontrollen und Verzweigungen

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

## Eingabe und Ausgabe

AREAD  
CSAVE  
CURSOR  
DATA  
GCURSOR  
GPRINT  
INPUT  
INPUT#  
INPUT#1  
LINE  
LOAD  
LPRINT  
PAUSE  
PRESET  
PRINT  
PRINT#  
PRINT#1  
PSET  
READ  
RESTORE  
USING  
WAIT

## Zuweisung und Deklarationen

CLEAR  
DIM  
LET

## Andere

BEEP	RADIAN
CLOSE	RANDOM
CLS	REM
CONSOLE	TROFF
GRAD	TRON
OPEN	

FUNKTIONEN

## Pseudovariablen

INKEY\$	OPEN\$
MEM	PI
MEM\$	POINT

## String-Funktionen

ASC	MID\$
CHR\$	RIGHT\$
LEFT\$	STR\$
LEN	VAL

## Numerische Funktionen

ABS	INT
ACS	LN
ASN	LOG
ATN	RND
COS	SGN
DEG	SIN
DMS	SQR
EXP	TAN

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. Das Kommando kann nur benutzt werden in Verbindung mit den Optionen CE-126P und CE-152.

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 PC 1350 im PROGRAMM- oder RUN-Modus, so wird das Programm von der Cassette in den Programmspeicher geladen. Ist der PC 1350 im RESERVE-MODUS, so werden die Daten in den Reservespeicher geladen. Sie müssen darauf achten, daß Sie keine Programme in den Reservespeicher und keine Reservedaten in den Programmspeicher laden.

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. Der Computer kann nicht entscheiden, ob die Daten für den Programm- oder den Reservespeicher gedacht sind. Werden Daten in den falschen Speicherbereich geladen, kann der Rechner nicht arbeiten. Sollte dieser Fehler gemacht worden sein, müssen Sie den RESET-Knopf an der Rückseite des Rechners drücken.
2. 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.
3. 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 der unteren Zeile ein Symbol "\*" angezeigt. Dieses Symbol verschwindet, wenn das Laden beendet ist. Wird ein Dateiname gesucht, erscheint dieses Symbol nicht, bevor der Ladevorgang beginnt.

Achtung: Das Symbol "\*" verschwindet normalerweise, wenn der Ladevorgang beendet ist. Es kann jedoch vorkommen, daß während der Ausführung des Befehls CLOAD in der vierten Zeile des Displays das Sternchen nicht verschwindet, wenn die Operation beendet ist, sondern in die dritte Zeile hochrutscht.

Das Bereitschaftssymbol (>) erscheint, wenn das Laden oder die Operation abgeschlossen ist.



## LOAD?

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. Es kann nur in Verbindung mit den Optionen CE-126P und CE-125 verwendet werden.

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 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 an dem unteren Rand des Displays, während das Programm überprüft wird. Am Ende des Prüfvorgangs verschwindet das Sternchen, und es erscheint das Bereitschafts-symbol.  
Wird das CLOAD?-Kommando in der vierten Zeile des Displays ausgeführt, verschwindet das Sternchen nicht, sondern taucht in der dritten Zeile des Bildschirms auf.

## 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 und GPRINT 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. Es kann nur in Verbindung mit den Optionen CE-126P und CE-152 benutzt werden.

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 dem Programm sowohl ein Dateiname als auch ein Passwort zugewiesen.

Im PROGRAMM- oder RUN-Modus wird der Inhalt des Programmspeichers, im Reserve-Modus der Inhalt des Reservespeichers auf Cassette abgespeichert.

Beispiel:

CSAVE "PRO3", "GEHEIM" Speichert das präsenste 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. Die Cursor-Spezifikationen bleiben erhalten.
7. Eine horizontale Bewegung des Cursors bei graphischen Darstellungen wird mit Ø angegeben. Die Einstellung für eine vertikale Bewegung gehört zu den Standardeinstellungen.
8. 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 ausdruck1,ausdruck2
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.

Anwendung:

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

Das LLIST-Kommando kann im PROGRAMM-Modus und im RUN-Modus benutzt werden.

Die erste Form druckt das gesamte Programm im Speicher aus. Die zweite Form druckt nur die mit ausdruck angegebene Zeile aus. Die dritte Form des Kommandos druckt das Programm 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.

Beispiel:

LLIST 100,200 Die Programmzeilen zwischen 100 und 200 werden ausgedruckt.

MERGE

- 1. MERGE
- 2. MERGE "dateiname"

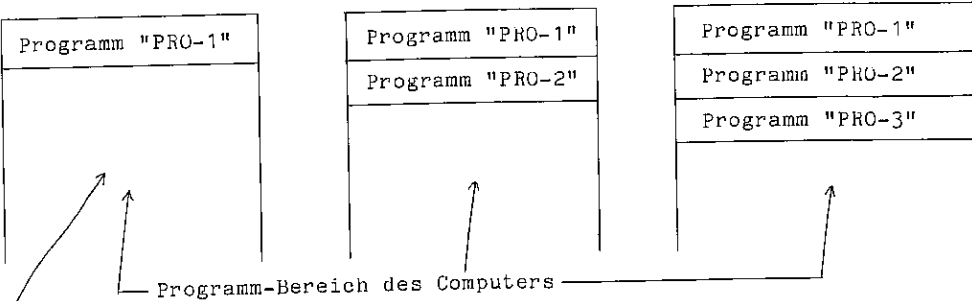
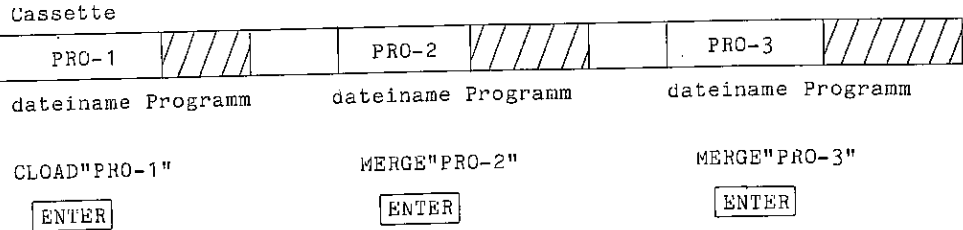
Abkürzung: MER., MERG.  
 Vergleiche: CLOAD

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 PC 1350 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.



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.

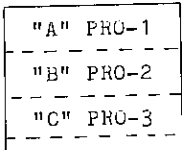
Sind im Speicher geschützte Programme und wird ein ungeschütztes Programm mit MERGE geladen, so wird dieses neu geladene Programm ebenfalls zu einem Passwort-geschützten Programm.



## KAPITEL 9

## 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 ausdrück, GOTO ausdrück) benutzen, um ein Programm zu starten, so wird immer PRO3 ausgeführt.



Benutzen Sie andererseits RUN "label" oder GOTO "label" oder eine andere definierte Taste, um den Ladevorgang zu starten, wird der Rechner nach dem angegebenen Label im Programm PRO3 suchen.

Wird das Label dort nicht gefunden, so setzt der Rechner seine Suche in dem Programm PRO1 fort. Ist die Suche auch dort nicht erfolgreich, wird PRO2 auf das gesuchte Label hin überprüft. Findet der Rechner das gewünschte Label, wird das Programm von der mit dem Label bezeichneten Zeile abgearbeitet.

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"  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 im Speicher bzw. Daten im Reservespeicher.

Anwendung:

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

Im Reserve-Modus wird der Reservespeicher gelöscht.

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 auszu-schalten, 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  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 "ABCDEFGH": A=123 .

Beispiel:

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

- \* Denken Sie an das Passwort, wenn Sie die RAM-Karte benutzen. Für den PC 1350 und die RAM-Karte existieren verschiedene Passworte. Wenn Sie die RAM-Karte einstecken, wird ein Passwort zugewiesen. Wenn Sie die RAM-Karte wieder entfernen, wird das Programm auf der RAM-Karte mit dem entsprechenden Passwort gespeichert und ist geschützt. Für den PC 1350 wird damit jedoch kein Passwort vergeben.
- Wird das Passwort erst vergeben, nachdem Sie die RAM-Karte entfernt haben, wird lediglich das Programm innerhalb des PC 1350 geschützt. Das Programm auf der RAM-Karte wird dann nicht mit dem Passwort geschützt. (Das zuvor für das Programm auf der RAM-Karte vergebene Passwort bleibt weiterhin gültig.)

## KAPITEL 9

## 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 ausdrück ausgeführt.

Für das Kommando RUN gilt 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. Die Cursor-Spezifikationen werden gelöscht.
7. Die Position des Cursors wird für graphische Darstellungen auf die Position (0,7) festgesetzt.
8. 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.

BEFEHLE

## AREAD

## 1. AREAD variablenname

Abkürzung: A., AR., AHE., AREA.  
 Vergleiche: Befehl INPUT und die Beschreibung der DEF -Taste in Kapitel 5.

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:

<pre>10 "X": AREAD N 20 PRINT N^2 30 END</pre>	Erstellen Sie sich dieses Beispiel-Programm im PROGRAMM-Modus.
--	--

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 PC 1350 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"	
20 BEEP 3	Erzeugt 3 Töne.
30 BEEP A	Erzeugt 5 Töne.
40 BEEP (A+4)/2	Erzeugt 4 Töne.
50 BEEP B\$	Nicht erlaubt, ERROR 9 wird ausgegeben.
60 BEEP -4	Es wird kein Ton erzeugt, aber auch keine Fehlermeldung ausgegeben.



## CHAIN

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

Abkürzung: CH., 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:

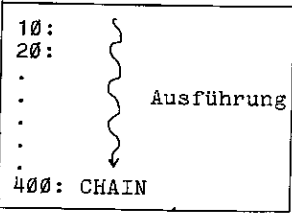
10 CHAIN                   Lädt das nächste Programm von Cassette und beginnt die Ausführung mit der niedrigsten Zeilennummer.

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.

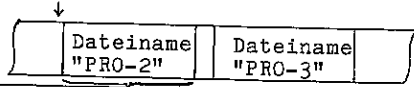
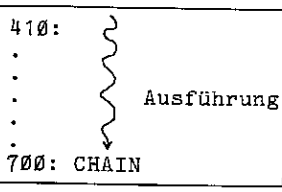
"PRO-1"



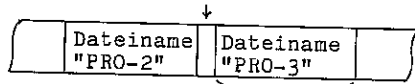
Cassettenrecorder  
 ("↓" markiert die Position des  
 Cassetten-Lese-Kopfes.)

400: CHAIN "PRO-2",410

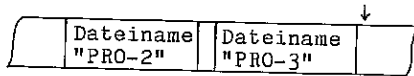
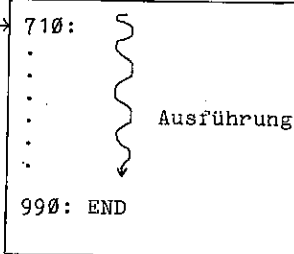
"PRO-2"



700: CHAIN "PRO-3",710



"PRO-3"



990: END

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

Abkürzung: 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 (siehe Kapitel 4). CLEAR setzt diese Variablen auf Null (numerische) bzw. auf leer (String).

Beispiele:

```
1Ø A=5: DIM C(5)
2Ø CLEAR
```

Setzt den C( ) zugewiesenen Speicher frei und A gleich Null.

## CLS

## 1. CLS

Vergleiche:     CURSOR

Wirkung:

Der CLS-Befehl löscht das Anzeigefeld.

Anwendung:

Der CLS-Befehl löscht den Bildschirm und versetzt das Display in die Startposition Ø.

Beispiel:

```
1Ø:WAIT 3
2Ø:INPUT A$
3Ø:FOR B=Ø TO 23
4Ø:CLS
5Ø:CURSOR B,1
6Ø:PRINT A$
7Ø:NEXT B
8Ø:CLS
9Ø:END
```

Dieses Beispiel-Programm bewegt das eingegebene Wort von der linken Seite der oberen Zeile zur rechten Seite der unteren Zeile. In jedem Durchgang der FOR/NEXT-Schleife wird die Anzeige mit dem CLS-Befehl gelöscht und die Cursor-Position mit dem CURSOR-Befehl verschoben. Der Inhalt von A\$ wird mit Hilfe des PRINT-Befehls angezeigt. Wenn Sie auf diese Weise Daten im Display anzeigen und löschen lassen, können Sie auch veranlassen, daß der Bildschirminhalt rollt.

CURSOR

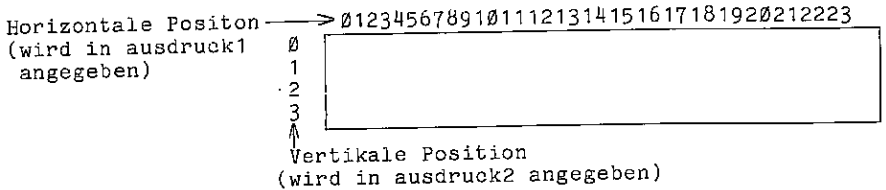
1. CURSOR ausdruck1,ausdruck2
2. CURSOR ausdruck
3. CURSOR

Abkürzung: CU., CUR., CURS., CURSO.  
 Vergleiche: GCURSOR, CLS, INPUT, PRINT, PAUSE

Wirkung:  
 Der CURSOR-Befehl markiert den Start-Punkt für die Bildschirm-Wiedergabe in Zeile und Spalte.

Anwendung:  
 Format 1 und 2 des Kommandos geben den Start-Punkt der Wiedergabe einer Ausgabe auf dem Bildschirm vor.

\* Format 1 spezifiziert sowohl die Spalte, als auch die Zeile.



Jede Position des Cursors am Bildschirm ist durch einen horizontalen und einen vertikalen Wert markiert. Der Wert des ausdruck1 in Form (1) und der Wert von ausdruck2 geben also die horizontale und die vertikale Position an.

ausdruck1 des Kommandos gibt die Spalte an. Der Wert darf nicht kleiner 0 (linker Bildschirmrand) und nicht größer als 23 (rechter Bildschirmrand) sein.

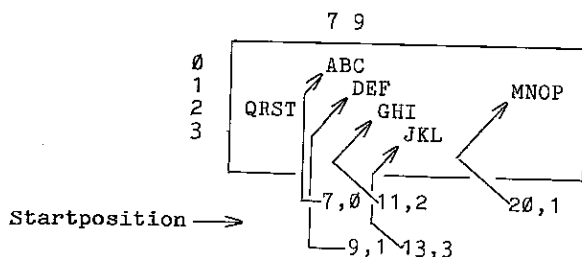
ausdruck2 des Kommandos gibt die Zeile an. Der Wert darf nicht kleiner als 0 (oberste Zeile) und nicht größer als 3 (unterste Zeile) sein.

Wird nach dem CURSOR-Kommando ein PRINT gegeben, so wird das erste Zeichen des PRINT-Kommandos an der durch den CURSOR-Befehl definierten Position angezeigt. Die folgenden Zeichen des PRINT-Kommandos werden dann dahintergesetzt.

Beispiel:

```
5 CLS
10 CURSOR 7,0:PRINT "ABC"
20 CURSOR 9,1:PRINT "DEF"
30 CURSOR 11,2:PRINT "GHI"
40 CURSOR 13,3:PRINT "JKL"
50 CURSOR 20,1:PRINT "MNOPQRST"
```

Bei Ausführung erhalten Sie folgendes Bild:



\* Format 2 des CURSOR-Kommandos definiert die Start-Position als Position des Cursors in einer bestimmten Spalte und einer bestimmten Zeile. Jede Position des Displays ist dabei fortlaufend numeriert.

#### Cursor-Positionen

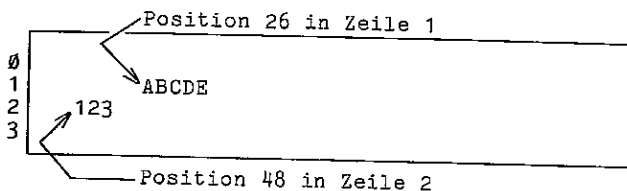
Zeile 0	"0" bis "23"
Zeile 1	"24" bis "47"
Zeile 2	"48" bis "70"
Zeile 3	"71" bis "95"

Die Spezifikation ausdrück innerhalb des Kommandos darf also nicht kleiner als 0 und nicht größer als 95 sein. Der ausdrück gibt den Start-Punkt einer Bildschirm-Wiedergabe, z. B. mit einem PRINT-Kommando, vor. Das erste Zeichen dieses Kommandos wird in die mit CURSOR vorgegebene Position gesetzt, die folgenden Zeichen des Wiedergabe-Kommandos direkt in aufsteigender Reihenfolge dahinter.

#### Beispiel:

```
5 CLS
10 CURSOR 48:PRINT 123
20 CURSOR 26:PRINT "ABCDE"
```

Folgendes Bild erhalten Sie bei Ausführung:



\* Format 3. des CURSOR-Kommandos löscht die Bildschirm-Wiedergabe-Start-Position.

Zeichen, die vor oder hinter einer mit CURSOR definierten Ausgabe stehen, bleiben erhalten. Sie können also mit dem CURSOR-Kommando und einem PRINT-Befehl einen Teil des Bildschirminhalts verändern, z. B. ein Wort oder eine Zahl. Um den Bildschirminhalt zu löschen, benutzen Sie das CLS-Kommando.

- \* Das CURSOR-Kommando kann auch in Verbindung mit dem INPUT-Kommando eingesetzt werden. Die Start-Position gibt dann den Punkt der Eingabe an. Nachdem das INPUT-Kommando ausgeführt ist, bleibt diese CURSOR-Position erhalten, d. h. ein nachfolgendes PRINT-Kommando würde die Eingabe überschreiben.

Beispiel:

```
5 CLS
10 CURSOR 0,2:GOTO 20
20 INPUT "DATEN=";A
30 PRINT A/2
```

Die in Zeile 10 definierte Position gibt sowohl den Start der Eingabe in Zeile 20 als auch den Start der Wiedergabe in Zeile 30 vor.

Beispiel:

```
5 CLS
10 CURSOR 0,2:GOTO 20
20 INPUT "DATEN=";A
25 CURSOR
30 PRINT A/2
```

Wenn Sie die Zeile 25 einfügen, so wird die CURSOR-Position für das PRINT-Kommando gelöscht.

- \* Wird eine Ausgabe länger gewählt, als sie auf dem Bildschirm dargestellt werden kann, so rollt der Bildschirm nach oben, um alle Zeichen wiedergeben zu können, selbst, wenn die Start-Position mit einem CURSOR-Kommando vorgegeben wurde.

Zwei Punkte sind zu beachten, wenn das CURSOR-Kommando in Verbindung mit einem INPUT-Kommando benutzt werden soll:

1. Zwischen INPUT und CURSOR muß ein GOTO-Kommando gesetzt werden.

```
10 CURSOR 5,2
20 GOTO 30
30 INPUT A
```

.  
.  
.

oder

```
10 CURSOR 5,2:GOTO 20
20 INPUT A
.
.
```

2. Achten Sie darauf, daß der Kommentar des INPUT-Kommandos nicht das Ende der vierten Zeile erreicht. Das Programm kann abgebrochen werden oder die Bildschirm-Wiedergabe verändert sich in unerwünschter Weise, unabhängig davon, ob ein GOTO eingesetzt wurde. Dann hilft nur noch die RESET-Taste auf der Rückseite des PC 1350. Er löscht alle Daten und Programme im Rechner und setzt den Prozessor auf seinen Start-Punkt zurück. Sie müssen das Programm anschließend neu laden.



## KAPITEL 9

## 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 eingerügt 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	Lädt die Werte aus den DATA-Zeilen
40: READ B(I)	in das Feld B( ). B(1) wird gleich 10,
50: PRINT B(I)	B(2) gleich 20, B(3) gleich 30 gesetzt,
60: NEXT I	usw.
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 PC 1350 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:

```
1Ø.DEGREE
2Ø:X=ASN 1
3Ø:PRINT x
```

X hat jetzt den Wert 9Ø, dem Arcussinus von 1.

## DIM

## 1. DIM dim liste

dim liste	dimensionierung oder dimensionierung,dim liste
dimensionierung	numerische dim oder string dim
numerische 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	größe oder größe,größe
größe	zahl der elemente
länge	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 Speicherraum für ein numerisches Feld mit 11 Elementen.
- 20: DIM C\$(4,4)\*10      Reserviert Speicherraum 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.

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 0 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 herauspringen. 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 dazugehörenden NEXT-Befehl in der Schleife liegen. Sie können bis zu fünf FOR/NEXT-Schleifen ineinanderlegen.

Beispiele:

<pre> 10:FOR I=1 TO 5 20:PRINT I 30:NEXT I </pre>	Diese Schleife gibt die Zahlen 1, 2, 3, 4 und 5 aus.
<pre> 40:FOR N=10 TO 0 STEP -1 50:PRINT N 60:NEXT N </pre>	Diese Schleife zählt rückwärts von 10 nach 0 und gibt die Schleifenvariable aus.
<pre> 70:FOR N= 1 TO 10 80:X=1 90:FOR F=1 TO N 100:X=X*F 110:NEXT F 120:PRINT X 130:NEXT N </pre>	Diese Schleife berechnet die Fakultäten der Zahlen von 1 bis 10 und gibt das Ergebnis auf der Anzeige wieder.

Achtung: Programm sollte nicht aus einer FO/NEXT-Schleife herauspringen. Es könnte hierbei passieren, daß Sie eine Fehlermeldung (ERROR 5) im Programmablauf (bei Programmen, die mehrere FOR/NEXT-Schleife enthalten) angezeigt bekommen.

## GOSUB

## 1. GOSUB ausdrück

Abkürzung: GOS., GOSU.

Vergleiche: GOTO, ON...GOTO, ON...GOSUB, 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 ausdrück 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 PC 1350 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 ausdrück 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 ausdrück 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

20:END

100:PRINT "HALLO"

110:RETURN

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.



## 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 eines Programms.

Anwendung:

Der Befehl veranlaßt den PC 1350, 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 Unterroutinensprüngen. 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
```

Wenn Sie ein "J" eingeben, gibt der Rechner das Wort JA aus, sonst das Wort NEIN.

## GRAD

## 1. GRAD

Abkürzung: GR., GRA.  
Vergleiche: DEGREE, RADIAN

Wirkung:

Mit dem GRAD-Befehl werden die Winkelwerte auf Neu-Grad umgeschaltet.

Anwendung:

Der PC 1350 hat drei Formen der Winkeldarstellung:

1. Alt-Grad
2. Radial-Wert
3. Neu-Grad

Die Argumente der Funktionen SINus, COSinus und TANgens können in dieser Form eingegeben bzw. die Ergebnisse der Umkehrfunktionen Arcus-Sinus, ArcusCoSinus und ArcusTanGens in diesen Formen ausgegeben werden.

Beispiel:

```
10:GRAD
20:X=ASN 1
30:PRINT X
```

X hat jetzt den Wert 100, den Gradienten von Arcussinus 1.

IF...THEN

1. IF bedingung THEN befehl
2. If bedingung befehl

Abkürzung: keine für IF : T., TH., THE.

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 Ausdruck, wie in Kapitel 4 beschrieben, sein. Es ist auch möglich, einen numerischen Ausdruck 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$
20:IF A$="JA" THEN GOTO 10
30:IF A$="NEIN" THEN GOTO 60
40:PRINT "JA ODER NEIN, BITTE"
50:GOTO 10
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, CURSOR, 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 oberen 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 Variablen 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.

## KAPITEL 9

Sollte der Befehl einen Prompt-String und die Variablenliste mehr als eine Variable enthalten, so wird der String nur bei der ersten Anforderung 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.

Wurde vor dem INPUT-Befehl ein CURSOR-Befehl gegeben, so wird das Fragezeichen bzw. der Prompt-String in der durch den CURSOR-Befehl gegebenen Position angezeigt.

Wurde direkt vor dem INPUT-Befehl ein PRINT-Befehl gegeben, der mit einem Semikolon (;) endet, so werden das Fragezeichen bzw. der Prompt-String und die eingegebenen Werte direkt hinter der Ausgabe des PRINT-Befehls 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 Anzeigefeld 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, 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.
```

Achtung: Wenn Sie Daten für eine Variable eingeben, beachten Sie, daß Sie nicht die Kapazität der Variable überschreiten. Wenn Sie Eingaben machen, die die Kapazität überschreiten, kann es vorkommen, daß bei Ausführung eines PRINT-Kommandos fehlerhafte Anzeigen auf dem Display erfolgen. Beseitigen Sie Fehler innerhalb eines INPUT-Kommandos durch Benutzung der **CLS**-Taste, und geben Sie dann die richtigen Daten ein. Wenn Sie den Fehler mit den Pfeiltasten beseitigen wollen, kann sich die Position der nachfolgend eingegebenen Daten auf dem Display verschieben. Trotzdem ist die Eingabe der Daten dann korrekt.

Die Ausführung des INPUT-Kommandos erfolgt nach Drücken der **ENTER**-Taste. Der Bildschirminhalt, der vor dem Drücken der **ENTER**-Taste angezeigt wurde, bleibt erhalten. Das Cursor-Symbol ( \_ oder  ) bleibt dann an unveränderter Position. Diese Cursor-Position bezieht sich jedoch nur auf die Bildschirm-Darstellung und betrifft nicht die eingegebenen Daten. Geben Sie ein GOTO-Kommando vor dem INPUT-Kommando ein (bei Programmen, die das INPUT-Kommando nach dem CURSOR-Kommando abarbeiten.)

**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 Variablenamen 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           Für die Zuweisung können einfache numerische
BI$="A" ENTER       oder String-Werte benutzt werden.
INPUT# AA,BI$ ENTER
```

## 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.



## KAPITEL 9

## LET

1. LET variable=ausdruck
2. variable=ausdruck

Abkürzung: LE.

Wirkung:

Der LET-Befehl weist Variablen Werte zu.

Anwendung:

Mit dem LET-Befehl weisen Sie der 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:

1Ø:I=1Ø	Weist I den Wert 1Ø zu.
2Ø:A=5*I	Weist A den Wert 5Ø zu.
3Ø:X\$=STR\$(A)	Weist X\$ den Wert "5Ø" zu.
4Ø:IF I>=1Ø THEN LET Y\$=X\$+".ØØ"	Weist Y\$ den Wert "5Ø.ØØ" zu.

## LPRINT

1. LPRINT druckausdr
2. LPRINT druckausdr,druckausdr,.....
3. LPRINT druckliste
4. LPRINT druckliste;
5. LPRINT

druckliste           druckausdr  
                          oder  
                          druckausdr;druckliste

druckausdr           ausdruck  
                          oder  
                          USING format;ausdruck

Das USING format wird unter USING beschrieben.

Abkürzung:           LP., LPR., LPRI., LPRIN.  
Vergleiche:         PAUSE, PRINT, USING, WAIT

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 angeschlossenen Drucker (siehe Seite 7-1). Um die Ausgabe des Programms auf dem Drucker durchzuführen, geben Sie ein CLOSE-Kommando ein.

Mit dem LPRINT-Befehl können Sie Ausgaben, die sonst auf der Anzeige des Rechners gemacht werden, mit dem Drucker ausdrucken lassen.

Die erste Form des Befehls druckt einen einzelnen Wert aus. Numerische Ausdrücke werden ganz rechts, String-Ausdrücke ganz links beginnend ausgedruckt.

Die zweite Form des Befehls beschränkt die Ausgabe der Druckspalten auf Gruppen mit je 12 Spalten. Die eingegebenen Werte werden in der Reihenfolge erster Wert erste Hälfte erste Zeile, zweiter Wert zweite Hälfte erste Zeile, dritter Wert erste Hälfte zweite Zeile, vierter Wert zweite Hälfte zweite Zeile ausgedruckt. Innerhalb der 12-Zeichen-Zeilen-Hälfte werden numerische Ausdrücke wieder ganz rechts und String-Ausdrücke ganz links ausgegeben.

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.

**Achtung:** Form 2 muß 2-8 Ausdrücke haben, sonst gibt es eine Fehlermeldung. Selbst, wenn die Wiedergabe-Startposition für die Form 2 durch Form 4 wiedergegeben ist, wird diese gelöscht und der Druck erfolgt wie oben beschrieben.

Form 3 druckt 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. Spalte in der Mitte eines numerischen Wertes, so erhalten Sie eine Fehlermeldung.

Form 4 des Befehls setzt mit dem Semikolon den Startpunkt des nächsten LPRINT-Befehls fest. Dieser beginnt direkt hinter dem zuletzt ausgegebenen Wert.

Die Form 5 des Befehls druckt nichts aus. Sie führt nur einen Zeilenvorschub aus.

#### 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$;A;B  
60:LPRINT  
70:LPRINT A*B;  
80:LPRINT Y$
```

## 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 Unter-routinen 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 Unter-routinen nach dem GOSUB ausgeführt. Ist der Wert 2, so wird die zweite ausgeführt usw. Nach dem RETURN aus der Unter-routine 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 Unter-routinen 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 PC 1350 kann nicht Kommata in Ausdrücken von Kommata zwischen Ausdrücken unterscheiden.

Beispiel:

```
10:INPUT A
20:ON A GOSUB 100,200,300
30:END
100:PRINT "ERSTE"
110:RETURN
200:PRINT "ZWEITE"
210:RETURN
300:PRINT "DRITTE"
310:RETURN
```

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.

## ON...GOTO

1. ON ausdruck GOTO ausdr liste

```
ausdr liste      ausdruck
                  oder
                  ausdruck,ausdr liste
```

Abkürzung: O.; G., GO., GOT.  
 Vergleiche: GOTO, GOSUB, 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 PC 1350 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
4. PAUSE druck liste;
5. PAUSE

druck liste       druck ausdr  
                   oder  
                   druck ausdr;druck liste

druck ausdr       ausdruck  
                   oder  
                   USING format;ausdruck

Das USING format wird unter USING beschrieben.

Abkürzung: PAU., PAUS.  
 Vergleiche: LPRINT, PRINT, CURSOR, 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 PC 1350 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. wurde die Startposition von einem USING-Befehl (Form 4) oder einem CURSOR-Befehl vorgegeben, beginnt die Ausgabe an dieser Stelle.

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.

Selbst, wenn die Startposition durch einen vorhergehenden PAUSE-Befehl der Form 4 oder durch einen CURSOR-Befehl vorgegeben wurde, werden die Werte in der Form 2 wie oben wiedergegeben.

- \* Die Anzahl der Werte in Form 2 muß zwischen 2 und 8 liegen.
- \* 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. Die Startposition kann mit einem CURSOR- oder USING-Befehl vorgegeben werden.

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.

Form 4 des Befehls gibt die eingegebene Liste an der linken Seite des Bildschirms wieder. Die auf das letzte Zeichen folgende Spalte ist die Startposition des nächsten Ausgabebefehls (z. B. PRINT):

Form 5 gibt den vorangegangenen Wert unverändert wieder.

Beispiele:

10:A=10:B=20:X\$="ABCDEF":Y\$="XYZ"

	<u>ANZEIGE</u>
20:PAUSE A	10.
30:PAUSE X\$	ABCDEF
40:PAUSE X\$,Y\$,A,B	ABCDEF      XYZ 10.                      20.
50:PAUSE Y\$;X\$;	XYZABCDEF
60:PAUSE A*B	XYZABCDEF200



## PRINT

1. PRINT druck ausdr
2. PRINT druck ausdr,....
3. PRINT druck liste
4. PRINT druck liste;
5. PRINT PRINT
6. PRINT = LPRINT
7. PRINT = PRINT

druck liste           druck ausdr

oder

druck ausdr;druck liste

druck ausdr           ausdruck

oder

USING format;ausdruck

Das USING format wird bei USING beschrieben.

Abkürzung:           P., PH., PRI., PRIN.  
 Vergleiche:         LPRINT, PAUSE, CURSOR, USING, WAIT

Wirkung:

Mit dem PRINT-Befehl können Ausgaben auf der Anzeige des PC 1350 oder der Option CE-126P gemacht werden.

Anwendung:

Mit dem PRINT-Befehl werden Texte 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 Startposition kann mit einem CURSOR- oder einem USING-Befehl (wie in Form 4) vorgegeben werden.

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 muß zwischen 2 und 8 liegen.
- \* 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. Die Startposition kann mit einem USING- (wie in Form 4) oder einem CURSOR-Befehl vorgegeben werden.

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.

Form 4 beginnt die Ausgabe links am Bildschirm. Die Startposition für einen folgenden Ausgabebefehl (z. B. PRINT) wird durch Form 4 des PRINT-Befehls auf die Spalte nach dem letzten Zeichen des letzten Wertes vorgegeben.

Achtung: Kombinieren Sie keinesfalls Bildschirmwiedergabe-Kommandos (PRINT etc.) mit Kommandos, die den seriellen E/A-Anschluß betreffen (LPRINT etc).

Die Formen 6 und 7 geben nichts aus. Die 6. Form setzt den PRINT-Befehl gleich dem LPRINT-Befehl. Alle Ausgaben werden jetzt also auf dem Drucker des CE-125 und nicht mehr auf der Anzeige des PC 1350 gemacht. Mit der 7. Form wird die 6. Form wieder zurückgesetzt, die Ausgabe erfolgt wieder über die Anzeige des PC 1350.

Beispiele:

10:A=123:B=5/):X\$="ABCDEF":Y\$="VWXYZ"

ANZEIGE

20:PRINT X\$,B	ABCDEF	5.55555E-01
30:PRINT A;B	123.5.5555555556E-01	
40:PRINT X\$;A	ABCDEF123	
50:PRINT Y\$;B	ABCDEF123.VWXYZ5.555555	56E-01

**PRINT#**

1. PRINT# var liste
2. PRINT# "dateiname";var liste

var liste            variable  
                               oder  
                               variable,var liste

Abkürzung:        P.#, PR.#, PRI.#, PRIN.#  
 Vergleiche:      INPUT#, PRINT, READ

Wirkung:

Mit dem PRINT#-Befehl werden Werte auf Cassette abgespeichert.

Anwendung:

Folgende Variablentypen 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.

## KAPITEL 9

## RADIAN

## 1. RADIAN

Abkürzung: RAD., RADI., RADIA.  
 Vergleiche: DEGREE, GRAD

Wirkung:

Der RADIAN-Befehl wechselt die Darstellung der Winkelwerte in die Radialform (Bogenmaß).

Anwendung:

Der PC 1350 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.  $2 \cdot \pi$ , 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 Zufallszahlen-generator.

Anwendung:

Wenn mit der RND-Funktion Zufallszahlen erzeugt werden, so startet der PC 1350 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 PC 1350 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=RND10
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)           Dimensionierung eines Feldes
20: WAIT 32
30: FOR I=1 TO 10      Zuweisung der Werte aus den DATA-Zeilen
40: READ B(I)          in die Feldvariablen B(0) bis B(10);
50: PRINT B(1)*2       B(0) ist 10, B(1) ist 20 usw.
60: NEXT I
70: DATA 10,20,30,40,50,60
80: DATA 70,80,90,100
90: END
```

## 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.

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( )	Alle Elemente haben den Wert 20.
60: PRINT B( )*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  
20:END  
100:PRINT "HALLO"  
110:RETURN
```

Das Programm druckt "HALLO" aus und springt zurück in Zeile 20.

## STOP

## 1. STOP

Abkürzung: S., ST., STO.  
Vergleiche: END; CONT-Kommando

Wirkung:

Mit dem STOP-Befehl wird die Ausführung eines Programms gestoppt.

Anwendung:

Erreicht der PC 1350 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 zur Folge, daß ein Programm wieder auf "normale" Weise ausgeführt wird.

Beispiel:

10:TRON  
20:FOR I=1 TO 3  
30:NEXT I  
40:TROFF

Bei Ausführung des Programms werden die  
Zeilennummern ausgegeben:  
10, 20, 30, 30 und 40

## 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 PC 1350 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   eingeben.

Nachdem ein Ergebnis an der im CURSOR-Kommando spezifizierten Stelle im Display angezeigt wurde, erscheint in der darauffolgenden Zeile die nächste Zeilennummer (vgl. auch CURSOR-Kommando).

Wenn im Trace-Modus nach einem CURSOR-Kommando die Start-Position spezifiziert wurde, wird diese gelöscht, wenn Variablen aufgerufen oder Berechnungen durchgeführt werden.

Beispiel:

10:TRON	Bei Ausführung des Programms werden die
20:FOR I=1 TO 3	Zeilennummern ausgegeben:
30:NEXT I	10, 20, 30, 30 und 40
40:TROFF	

## USING

1. USING
2. USING "spezifikation"
3. USING zeichen variable

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 Ausgabebefehle 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 zusammzusetzen:

- # 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"

20:PRINT USING "##.##^";A

30:PRINT USING "#####";X\$

40:PRINT USING "###&amp;&amp;";A;X\$

ANZEIGE

1.25E02

ABCDEF

125ABC

WAIT

1. WAIT
2. WAIT ausdruck

Abkürzung: W., WA., WAI.  
Vergleiche: PAUSE, PRINT

Wirkung:

Der WAIT-Befehl setzt die Zeit fest, die eine Ausgabe auf der Anzeige des PC 1350 wiedergegeben werden soll.

Anwendung:

Bei normaler Ausführung eines Programms wartet der PC 1350 nach einem PRINT-, GRPINT-, PSET-, PRESET- oder LINE-Befehl, bis die **ENTER** - Taste gedrückt wird. Der WAIT-Befehl weist den PC 1350 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 1/64 Sekunde. WAIT ist zu schnell, als daß die Ausgabe gelesen werden könnte; WAIT 65535 setzt das Intervall auf ca. 17 Minuten. WAIT ohne Ausdruck setzt den PC 1350 auf seine normale Arbeitsweise zurück.

Beispiel:

10:WAIT 64

Der Rechner wartet nach PRINT etwa 1 Sekunde.



FUNKTIONEN

Pseudovariablen sind eine Gruppe von Funktionen, die ohne Argument wie einfache Variablen behandelt werden.

## INKEY\$

## 1. INKEY\$

INKEY\$ ist eine String-Pseudovariablen, die den Wert der letzten, auf der Tastatur gedrückten Taste zugewiesen bekommt. Die Tasten ENTER, CL, SHIFT, DEF, MODE, INS, DEL, SML, ↑-Pfeil, ↓-Pfeil, ▶-Pfeil und ◀-Pfeil haben den Wert NUL (=leer). mit INKEY\$ können Eingaben über die Tastatur ohne ein abschließendes ENTER gemacht werden.

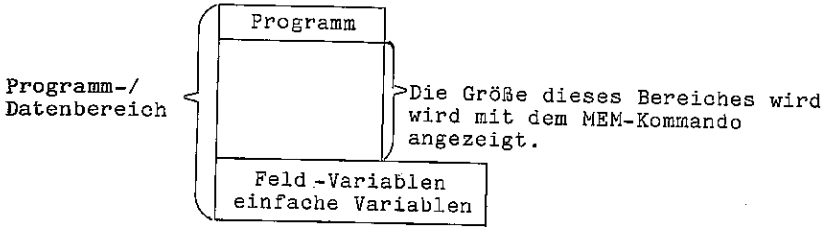
```
10:A$= INKEY$
20:B=ASC A$
30:IF B=0 THEN GOTO 10
40:IF B=...
```

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.

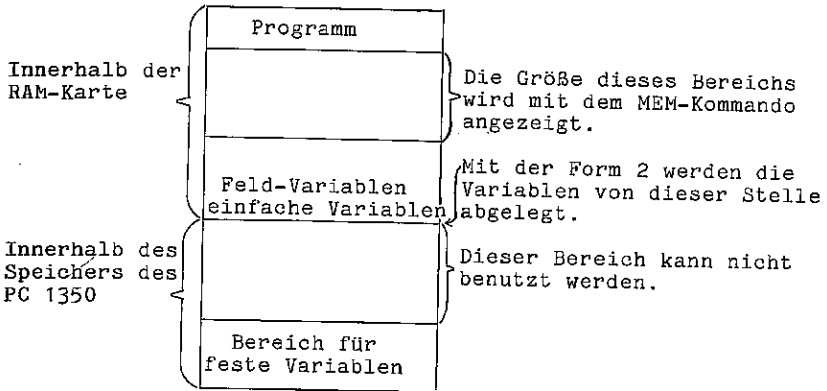
MEM

1. MEM
2. MEM "C"
3. MEM "B"

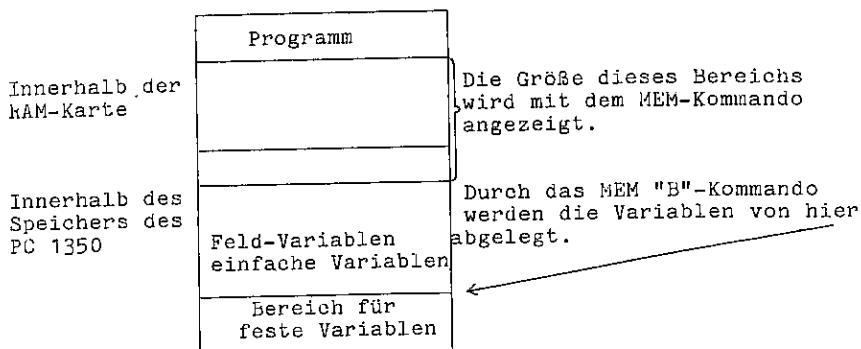
Mit der Form 1 erhalten Sie eine Angabe über die Zahl der freien Bytes im Programm-/Datenbereich (Bereich, der nicht durch ein Programm, Array-Variablen oder einfache Variablen belegt ist).



Wenn Sie eine RAM-Karte benutzen, können Sie Form 2 so definieren, daß Variablen auf der RAM-Karte gespeichert werden.



Mit Form 3 übertragen Sie die Startposition des Speicherbereichs der Variablen, wie Sie ihn auf der RAM-Karte mit Form 2 definiert haben, in den Speicher des PC 1350.



Wenn Sie die Speicherplatz-Startposition der Variablen in Form 2 oder Form 3 ändern, machen Sie die zuvor gespeicherten Variablen für den weiteren Gebrauch unbrauchbar.

- \* Wenn Sie die RAM-Karte nicht benutzen, erhalten Sie bei Eingabe des Kommandos MEM "C" die Fehlermeldung ERROR 1.
- \* Die Werteszuschreibung des MEM "C"-Kommandos wird auf der RAM-Karte gespeichert. Entfernen Sie die RAM-Karte aus dem PC 1350, gehen diese Daten verloren. (Der Rechner liest dann das MEM "B"-Kommando.)  
Setzen Sie die RAM-Karte erneut ein, erhalten Sie wieder den Wert aus dem MEM "C"-Kommando.

1. Die Programmgröße (in bytes) kann durch folgende Eingaben angezeigt werden:

Beispiel: PC 1350 ohne RAM-Erweiterungen

RUN-Modus

CLEAR

(löscht die einfachen Variablen,  
die Feld-Variablen usw.)

3070-MEM

(Es wird die Programmgröße  
- in Bytes - angezeigt.)

↑  
Eingabe der gesamten RAM-Speichergröße (in Bytes).

## RAM-Speichergröße (in Bytes)

PC 1350 ohne RAM-Erweiterung	PC 1350 + CE-201M	CE-210M	PC 1350 + CE-202M	CE-202M
3070	11 262	8142*	19 454	16 334*

\* Größe der RAM-Erweiterung wird angezeigt, nachdem das Kommando MEM "C" gegeben wurde.

2. Prüfen, ob ein Programm in der RAM-Erweiterung abgespeichert ist.

Ein in die RAM-Erweiterung geschriebenes Programm, welches die Größe der RAM-Erweiterung überschreitet, wird zerstört, falls Sie die RAM-Erweiterung aus dem PC 1350 entfernen.

Deshalb überprüfen Sie die Programmgröße nach folgendem Schema:

a) Das Kommando MEM "B" wurde gegeben.

Geben Sie ein

MEM

und prüfen Sie, ob der Wert größer oder gleich 3120 ist.

Hinweis: 3120 = Programm-/Daten-Bereich innerhalb des PC 1350  
(3070) + Systembereich (50).

b) Das Kommando MEM "C" wurde gegeben.

Geben Sie ein

MEM\$

ein und achten Sie darauf, daß "C" angezeigt wird.

\* Sollte ein Programm geschrieben sein, welches die Größe der RAM-Erweiterung überschreitet, nachdem das Kommando MEM "C" gegeben wurde, so wird MEM "C" gelöscht und das Kommando MEM "B" gegeben.

Falls "B" angezeigt wird, nachdem Sie MEM\$  eingegeben haben, so geben Sie nun

3120 - MEM

ein, und kürzen das Programm um den Wert in Bytes, der angezeigt wird.

MEM\$

1. MEM\$

Wirkung:

Mit Hilfe dieses Kommandos können Sie sich anzeigen lassen, ob Sie sich derzeit im Speicherbereich der RAM-Erweiterung oder im Speicherbereich des PC 1350 befinden.

Beispiel:

MEM\$

Befinden Sie sich zur Zeit im Speicherbereich des PC 1350 (MEM "B"-Kommando) wird "B" angezeigt, andernfalls (MEM "C"-Kommando) wird "C" angezeigt.

\* "B" wird angezeigt, falls keine RAM-Erweiterung installiert ist.

PI

## 1. PI

PI ist eine numerische Pseudovariablen, die den Wert  $\pi$  hat. Sie hat die gleiche Funktion wie die spezielle Taste  $\boxed{\pi}$  auf der Tastatur. Wie alle anderen Zahlen wird auch der Wert von PI mit einer Genauigkeit von 10 Stellen wiedergegeben (3.141592654).

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. Der PC 1350 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 Arcuscossinus des Arguments. Arcuscossinus ist die Umkehrfunktion des Cosinus. Das Ergebnis ist also der Winkel zu dem angegebenen Wert. Die Darstellung des Ergebnisses ist abhängig vom Winkelmodus des PC 1350 (DEG, RAD, GRAD).

ACS .5 ist im DEG-Modus gleich 60

ASN

1. ASN numerischer ausdruck

Errechnet den Arcussinus des angegebenen Arguments. Arcussinus ist die Umkehrfunktion des Sinus. Die Darstellung des Ergebnisses ist abhängig vom Winkelmodus des PC 1350 (DEG, RAD, GRAD).

ACS .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 PC 1350 (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 PC 1350 (DEG, RAD, GRAD).

COS 60 ist im DEG-Modus gleich .5

## 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 Gradzahl, 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" werden 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

INT

1. INT numerischer ausdruck

Das Ergebnis ist der ganzzahlige Anteil des Arguments.

INT PI ist gleich 3.

LOG

1. LOG numerischer ausdruck

Errechnet den Logarithmus zur Basis 10 des Arguments.

LOG 100 ist gleich 2.

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-----	
	Untergrenze	Obergrenze
.5	0<	<1
2	1	2
2.5	1	3

Die Sequenz ist jedes Mal die gleiche, wenn der PC 1350 eingeschaltet wird, da die Basiszahl immer die gleiche ist. Um diese Basis-Zahl zufällig zu ändern, brauchen Sie den RANDOM-Befehl.

## 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 PC 1350 (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  $\sqrt{\quad}$ -Taste der Tastatur.

SQR 4 ist gleich 2.

## TAN

## 1. TAN numerischer ausdruck

Errechnet den Tangens des Winkelarguments. Die Darstellung ist abhängig vom Winkelmodus des PC 1350 (DEG, RAD, GRAD).

TAN 45 ist im DEG-Modus gleich 1.

STRING-FUNKTIONEN

Ein String ist eine Kette aus alphanumerischen und Sonderzeichen. String-Funktionen sind eine Gruppe von Operationen, die Strings manipulieren. Sie nehmen ein 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-Dialekten muß das Argument der Funktion in Klammern gesetzt werden. Nicht so bei dem PC 1350. 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 PC 1350 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, numasdr2)

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.

GRAPHIK-KOMMANDOS

## GCURSOR

1. GCURSOR (ausdruck1,ausdruck2)

Abkürzung: GC., GCU., GCUR., GCURS., GCURSO.

Vergleiche: CURSOR, GPRINT

Wirkung:

Definiert den Startpunkt einer Bildschirmausgabe als Bildschirmpunkt.

Anwendung:

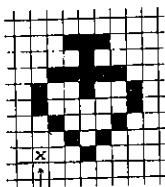
Das GCURSOR-Kommando gibt den Startpunkt für das GPRINT-Kommando. Der Graphik-Cursor wird an die angegebene Position gesetzt.

\* Der Bildschirm ist aus 32 Zeilen mit 150 Punkten aufgebaut. Jeder dieser Punkte kann in einem X,Y-Koordinatensystem angesprochen werden. X darf dabei nicht kleiner als 0 und nicht größer als 149, Y nicht kleiner als 0 und nicht größer als 31 sein. ausdruck1 des Kommandos gibt den X-Wert, ausdruck2 den Y-Wert an.

Beispiel:

```
10 GCURSOR (71,20)
20 GPRINT "1824458F452418"
```

Bei der Ausführung wird folgendes Bild in der Mitte des Bildschirms wiedergegeben. Das "x" gibt die durch das GCURSOR-Kommando vorgegebene Startposition.



Das "x" stellt die Position (71,20) dar.

**Achtung:** Der Wert des ausdruck1 und ausdruck2 kann zwischen -32768 und +32767 liegen. werte, die kleiner 0 oder größer 149 für ausdruck1, bzw. größer als 31 für ausdruck2 sind, setzen den Cursor an eine Position außerhalb des Bildschirms, so daß eine Graphik nicht dargestellt wird.

\* Der Graphik-Cursor wird automatisch auf (0,7) gesetzt, wenn der PC 1350 ein RUN- oder ein CLS-Kommando erhält oder die Tasten **SHIFT** und **CA** gedrückt werden.

Wird ein Programm mit GOTO gestartet, so bleibt der Y-Wert der Cursor-Position erhalten. Der X-Wert wird auf 0 zurückgesetzt.

## GPRINT

1. GPRINT string
2. GPRINT ausdruck;ausdruck;ausdruck;...
3. GPRINT

Abkürzung: GP., GPR., GPRI., GPRIN.  
 Vergleiche: GCURSOR, PRINT

Wirkung:

Gibt die angegebenen Graphik-Symbole aus.

Anwendung:

Das GPRINT-Kommando gibt die angegebenen Graphik-Symbole auf dem Bildschirm aus. Ein Graphik-Symbol besteht aus einer senkrechten Linie von 8 Bildschirmpunkten.

- \* Form 1 des Kommandos teilt dieses Symbol in obere und untere Hälfte zu je 4 Punkten. Sie werden als String (Zeichenkette) in Anführungsstrichen (" ") im GPRINT-Kommando angegeben. Die Hälften werden als hexadezimale Zahlen (0 bis F) eingesetzt.

0	4	B	C
1	5	9	D
2	6	A	E
3	7	B	F

Jedes dieser Paare repräsentiert ein Graphik-Symbol mit 8 vertikalen Punkten. Eine einzelne Ziffer am Ende wird ignoriert.

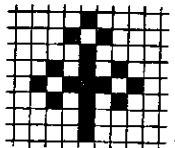
Beispiel:

GPRINT "102812FD122810"

x\_x\_x\_x\_x\_x\_x\_x

Das "X" steht hier für die obere Hälfte, das "\_" für die untere Hälfte des Symbols.

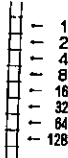
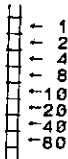
Das Ergebnis sieht also so aus:



} Die oberen vier Punkte  
 } Die unteren vier Punkte

0820280 — die obere,  
 121F121 — die untere Hälfte im Hexadezimal-Code.

\* In Form 2 des Kommandos wird die Graphik als vertikale 8-Punkt-Sequenz in dezimaler oder hexadezimaler Form eingegeben. Der Wert eines Punktes dieser Sequenz wird aus der folgenden Illustration ersichtlich.



Die Form einer Sequenz wird durch die Summe der Werte der einzelnen Punkte bestimmt. Ein nicht gesetzter Punkt hat natürlich den Wert Null.

Beispiel:

\* Hexadezimale Definition einer Punkt-Graphik:

```
GPRINT &10;&28;&12;&FD;&12;&28;&10
```

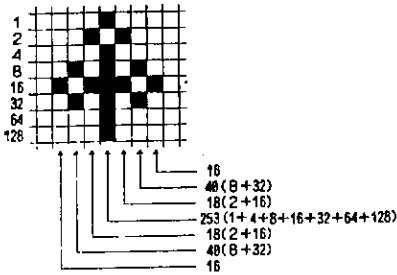
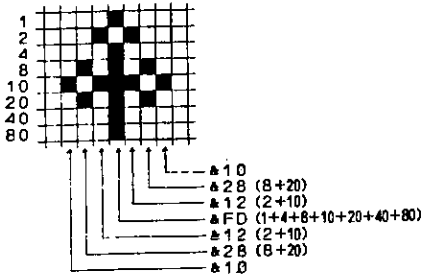
\* Dezimale Definition einer Punkt-Graphik:

```
GPRINT 16;40;18;253;18;40;16
```

Das Ergebnis ist für beide Formen das gleiche, nämlich die o. a. Graphik. Die Werte errechnen sich aus der Summe der gesetzten Punkte pro Spalte, also:

Spalte	Punkte	HEX	DEZ
1	5	&10	16
2	4/6	8+20=&28	8+32=40
3	2/5	2+10=&12	2+16=18
4	1/3/4/5/6/7/8	1+4+8+10+20+40+80=&FD	1+4+8+16+32+64+128=253
5	2/5	2+10=&12	2+16=18
6	4/6	8+20=&28	8+32=40
7	5	&10	16





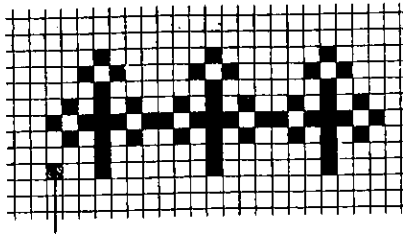
\* Form 3 des Kommandos erhält die auf dem Bildschirm dargestellte Graphik.

\* Wurde der Cursor mit GCURSOR vorher an eine bestimmte Stelle des Bildschirms gesetzt, so nimmt der achte Punkt der ersten Spalte des Bildes diese Position an. Die folgenden Bildspalten nehmen die folgenden Bildschirmspalten ein.

Beispiel:

```

10 AA$="102812FD122810"
20 GCURSOR (60,20)
30 GRPINT AA$;AA$;AA$
  
```



Der achte Punkt der ersten Wiedergabe des Bildschirms nimmt die Position (60,20) - durch das G\_CURSOR-Kommando vorgegeben - ein.

Achtung: Endet das G\_PRINT-Kommando mit einem Semikolon (;), so ist die folgende Spalte die neue Cursor-Spalte. Endet G\_PRINT mit einem Doppelpunkt (:) oder ENTER, so wird die G\_CURSOR-Spalte gleich Null gesetzt.

## LINE

1. LINE (A)-(B),C,D,E

(A)	(ausdruck1, ausdruck2)
(B)	(ausdruck3, ausdruck4)
C	S oder R oder X
D	ausdruck5
E	B oder BF

Abkürzung: LIN.  
Vergleiche: GCURSOR, PSET

Wirkung:  
Zieht eine Linie zwischen zwei Punkten.

Anwendung:  
Zwischen den beiden Punkten (ausdruck1, ausdruck2) und (ausdruck3, ausdruck4) wird eine Linie gezogen.

Beispiel:

LINE (0,0)-(149,31)

Zieht eine Linie von der oberen linken zur unteren rechten Ecke des Bildschirms.

\* Die Werte von ausdruck1 bis ausdruck4 können aus dem Bereich von -32768 bis +32767 liegen. Allerdings können nur die Werte von 0 bis 149 für ausdruck1 und ausdruck3, und 0 bis 31 für ausdruck2 und ausdruck4 auf dem Bildschirm dargestellt werden.

Wird ein Punkt außerhalb des Bildschirms gesetzt, erhalten Sie dennoch keine Fehlermeldung. Der Versuch, einen Punkt außerhalb des Bereichs von -32768 bis +32767 zu setzen, resultiert in der Fehlermeldung ERROR 3.

\* Term (A) kann weggelassen werden. Die Linie wird dann entweder von (0,0) oder, wenn vorher bereits ein LINE-Kommando gegeben wurde, von der vorangegangenen Endposition aus gezogen.

Beispiel:

```
5 CLS
10 LINE (10,0)-(149,16)
20 WAIT:LINE -(70,31)
```

Achtung: Da der Bildschirm aus einzelnen Punkte aufgebaut ist, werden Diagonale und Kurven nicht sauber durchgezogen.

\* Term C gibt an, auf welche Weise ein Punkt gesetzt werden soll:

- S: Setzt die Punkte der gegebenen Linie.  
 R: Löscht die Punkte der gegebenen Linie, die Möglichkeit, in einem gesetzten Feld eine helle Linie zu ziehen.  
 X: Setzt einen Punkt der Linie, wenn er vorher noch nicht gesetzt war und löscht ihn, wenn er gesetzt war.

\* Term D setzt den "Typ" der Linie, die gezogen werden soll.

Wenn der Wert von `ausdruck5` zum Beispiel gleich DEZ 26214 (&6666) ist, so wird eine Linie mit folgendem Muster erzeugt:



16 Punkte; die definierte Zeile wird gebildet, indem die links gezeigte Sequenz wiederholt wird.

Die Zahl 26214 (HEX &6666) kann als Binärzahl wiedergegeben werden:

0110011001100110

Wenn Sie das Punktmuster von oben mit dieser Binärzahl vergleichen, werden Sie erkennen, daß ungesetzte Punkte den Nullen und gesetzte den Einsen entsprechen. Eine Binärzahl gibt also direkt das Muster der Linie wieder, das mit der Dezimal- (oder Hexadezimal-)Zahl gesetzt wurde. `ausdruck5` gibt immer die Umsetzung einer 16stelligen Binärzahl wieder. Die Linie erscheint also nicht, wenn der Wert gleich 0 ist. 65535 (&FFFF) definiert alle Punkte, so daß die Linie durchgezogen erscheint. Wird Term D weggelassen, so wird 65536 (&FFFF) gesetzt. In Term C gesetzte Optionen werden dennoch ausgeführt, so daß ein gesetztes Punktmuster u. U. ganz oder teilweise invertiert ausgegeben wird.

- \* Der Wertebereich für Term D liegt zwischen 0 und 65536 (&FFFF).
- \* Term E definiert die Linie als Diagonale eines Rechtecks, deren Anfangs- und Endpunkt mit den Termen (A) und (B) gesetzt sind.

B: Definiert die Diagonale eines Rechtecks.  
 BF: Definiert die Linie als Diagonale eines Rechtecks und füllt es aus (malt es aus).

#### Beispiel:

```

10 CLS:WAIT 0
20 AA$="102812FD122810"
30 GCURSOR (64,20)
40 GPRINT AA$;AA$;AA$
50 LINE (24,0)-(124,31),&F18F,B
60 LINE (34,3)-(114,28),X,BF
70 GOTO 60
  
```

## POINT

1. POINT (ausdruck1,ausdruck2)

Abkürzung: POI., POIN.

Vergleiche: GCURSOR, PSET, PRESET

Wirkung:

Ermittelt den Status des angegebenen Punktes.

Anwendung:

Ist der mit (ausdruck1,ausdruck2) definierte Punkt gesetzt, so wird eine "1" zurückgegeben. Ist der Punkt nicht gesetzt, eine "0". Liegt der Punkt außerhalb des Bildschirms, eine "-1".

\* Der Wert für ausdruck1 und ausdruck2 muß im Bereich von -32768 und +32767 liegen. Sollen die Punkte auf dem Bildschirm sein, darf der Wert von ausdruck1 die Grenzen von 0 bis 149 und ausdruck2 von 0 bis 31 nicht überschreiten.

Beispiel:

10 CLS:WAIT 0:A=75	
20 LINE (50,0)-(50,31)	← Zwei vertikale Linien werden gezogen.
30 LINE (100,0)-(100,31)	←
40 PSET (A,16)	Setzt einen Punkt.
50 B= POINT (A+1,16)	Prüft, ob der nächste Punkt rechts gesetzt ist.
60 IF B THEN 150	Wenn gesetzt, springe nach Zeile 150.
70 PSET (A+1,16)	Setzt einen Punkt in der nächsten Position rechts.
80 PRESET (A,16)	Löscht den alten Punkt.
90 A=A+1	Erhöht A um 1.
100 GOTO 50	Rücksprung nach Zeile 50.
150 B= POINT (A-1,16)	Prüft, ob der nächste Punkt links gesetzt ist.
160 IF B THEN 50	Wenn gesetzt, springe nach Zeile 50.
170 PSET (A-1,16)	Setzt einen Punkt in der nächsten Position links.
180 PRESET (A,16)	Löscht den alten Punkt.
190 A=A-1	Erniedrigt A um 1.
200 GOTO 150	Rücksprung nach Zeile 150.

Dieses Programm bewegt einen Punkt vor und zurück zwischen zwei angezeigten vertikalen Linien.

## PRESET

1. PRESET (ausdruck1,ausdruck2)

Abkürzung: PRE., PRES., PRESE.  
 Vergleiche: PSET, GCURSOR, POINT

Wirkung:

Löscht den definierten Punkt auf dem Bildschirm.

Anwendung:

Der mit (ausdruck1,ausdruck2) definierte Punkt wird gelöscht. Die Werte von ausdruck1 und ausdruck2 müssen im Bereich von -32768 und +32767 liegen. Der Bildschirm stellt aber nur die Werte von 0 bis 149 für ausdruck1 und 0 bis 31 für ausdruck2 dar. Alle anderen Punktkoordinaten liegen außerhalb des Bildschirms.

Beispiel:

```
10 CLS:WAIT 0
20 LINE (20,0)-(130,31),BF
30 FOR X=-25 TO 25 STEP .5
40 Y=-1* SQR ABS (25*25-X*X)
50 PRESET (X+75,Y+31)
60 NEXT X
70 WAIT:GPRINT
```

Das Programm erzeugt einen Halbkreis in einem gefüllten Rechteck.

## PSET

1. PSET (ausdruck1,ausdruck2)
2. PSET (ausdruck1,ausdruck2),X

Abkürzung: PS., PSE.  
 Vergleiche: PRESET, GCURSOR, POINT

Wirkung:  
 Setzt oder invertiert einen Punkt auf dem Bildschirm.

- Anwendung:
- \* Form 1 setzt einen Punkt, der mit (ausdruck1,ausdruck2) definiert wird.
  - \* Form 2 löscht den mit (ausdruck1,ausdruck2) definierten Punkt, wenn dieser bereits gesetzt ist bzw. setzt den Punkt, wenn er noch nicht gesetzt war.
  - \* Der wert für ausdruck1 und ausdruck2 muß im Bereich von -32768 bis +32767 liegen. Der Bildschirm gibt aber nur die Punkte mit den Koordinaten für ausdruck1 von 0 bis 149 und für ausdruck2 von 0 bis 31 wieder.

Beispiel:

```
10 CLS:WAIT 0:DEGREE
20 FOR A=0 TO 600
30 B=-1*SIN A
40 Y=INT (B*16)+16
50 X=INT (A/4)
60 PSET (X,Y)
70 NEXT A
80 WAIT:GPRINT
```

Dieses Programm erzeugt eine Sinuskurve auf dem Bildschirm.

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 40 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 PC 1350 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 Eingabedaten "CR" (Kontrollcode ØDH) 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.

## 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:

Das 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 in den PC 1350 ein:

```
10 OPEN
100 REM **ABC-12**
65279 END
```

Geben Sie dann das Kommando:

```
LLIST 
```

Das Listing wird auf dem Drucker ausgegeben. Geben Sie jetzt das Kommando:

```
OPEN 
LLIST 
```

Das Listing wird jetzt über das Interface ausgegeben.

- \* Form 2 sendet nur die Zeile, die mit dem Wert von `ausdruck` oder dem eingegebenen "label" gekennzeichnet ist.
- \* Form 3 sendet alle Zeilen, die zwischen den mit `ausdruck1` und `ausdruck2` benannten Zeilen liegen (inklusive). `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 in den PC 1350 eingeladen, so wird mit LLIST nur das letzte geladene Programm ausgegeben. Wollen Sie die vorher geladenen Programme ausgeben 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.

## 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 PC 1350 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 den Textendecode.
  2. Oder drücken Sie die BRK-Taste des PC 1350, um die Übertragung zu beenden.
- \* Der Reservespeicherbereich kann nicht über das serielle Interface gesendet werden.

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:

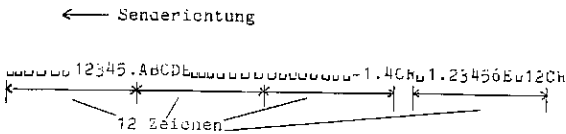
Wurde die Leitung zum seriellen Interface mit dem OPEN-Kommando geöffnet, so werden die Daten im ASCII-Code übertragen. Ist die Leitung geschlossen, werden die Daten auf dem Drucker ausgegeben.

- \* Form 1 sendet den Wert des Ausdrucks oder die Zeichenkette (String). Ist der Wert des Ausdrucks negativ, wird ihm ein Minuszeichen (-) vorangesetzt. 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 vorangesetzt.

- \* Form 3 verändert nicht die Länge der zu sendenden Daten. Einer positiven Zahl wird kein Leerfeld vorangestellt.

Beispiel:

```
50 LPRINT -123;"ABC";%&/.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.
- \* 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 und 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 PC 1350 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

Es können nur Zeichen im ASCII-Code gesendet oder empfangen werden. Deshalb ist A immer gesetzt.

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 oder der Rechner abgeschaltet 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\$ 

Ausgabe:

1200,N,8,1,A,C,&amp;1A

PRINT# 1

1. PRINT# 1 variable,variable,variable, ...

Abkürzung: P.#1, PR.#1, PRI.#1, PRIN.#1  
 Vergleiche: OPEN, INPUT#1

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.

\* 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: 5Ø PRINT #1A,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

PKINT# 1A,B\$

in dieser Form gesendet (Endcode sei CR):

1.2345ØØØØØØE04"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):  $C(\emptyset, \emptyset) \Rightarrow C(\emptyset, 1) \Rightarrow C(\emptyset, 2) \Rightarrow C(\emptyset, 3) \Rightarrow C(1, \emptyset) \dots$

Achtung:

- \* Erweiterte Variable, wie A(27) oder darüber sowie einfache und Feld-Variable, 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 PC 1350 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 PC 1350 über das serielle Ein-/Ausgabe-Interface.

Anwendung:

Wenn die Leitung zum seriellen Interface mit dem OPEN-Kommando geöffnet ist, werden Programme im ASCII-Code übertragen. Ist die Leitung geschlossen, wird das Kommando ignoriert.

- \* An das Ende des Programms wird der Textendecode angehängt.
- \* Das SAVE-Kommando wird nicht ausgeführt, wenn ein Passwort gesetzt wurde.

## TEXT-FUNKTIONSKOMMANDOS

## BASIC

## 1. BASIC

Abkürzung: BA., BAS., BASI.  
Vergleiche: TEXT

Wirkung:

Löscht den TEXT-Modus, nur in der Direkteingabe im PROGRAMM-Modus erlaubt.

Anwendung:

Die Ausführung des BASIC-Kommandos löscht den TEXT-Modus und schaltet den PC 1350 in den Basic-Modus zurück.

Das Bereitschaftssymbol wechselt von "<" nach ">".

\* Der Wechsel aus dem TEXT- in den BASIC-Modus wandelt einen Text im PC 1350 in ein Programm um (Interner Code).

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 PC 1350 vorgesehen sind, können nicht ausgeführt werden.

\* Während der Umformung wird ein Stern (\*) am rechten Ende der vierten Zeile 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 PC 1350 in den TEXT-Modus, nur erlaubt in Direkteingabe im PROGRAMM-Modus.

Anwendung:

Die TEXT-Funktion (oder -Modus) wird gebraucht, wenn ein Programm in den PC 1350 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 PC 1350 in den TEXT-Modus. Daten, bestehend aus einer Zahl, der Zeilennummer und die Programminformation, Befehle 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 PC 1350 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. "1." für "INPUT", so wie sie eingegeben werden, angezeigt und gespeichert.

- \* Ein Programm, das im internen Code des PC 1350 gespeichert ist, wird bei Umschaltung in den TEXT-Modus in den ASCII-Code umgewandelt.
- \* Während der Programmumwandlung wird in der vierten Zeile 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.



## KAPITEL 10

PANNENHILFE

In diesem Kapitel sollen Sie einige Hinweise erhalten, was Sie unternehmen können, wenn Ihr SHARP PC 1350 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.

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.

Die Anzeige zwar funktioniert, aber keine Reaktion auf Tastendruck erfolgt.

- 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.

Sie eine Rechnung oder Antwort eingeben, aber keine Reaktion erfolgt.

- 1) Drücken Sie **ENTER** .

Sie ein BASIC-Programm abarbeiten lassen, etwas angezeigt wird und das Programm dann anhält.

- 1) Drücken Sie **ENTER** .

Sie eine Rechnung eingeben und diese in der Form eines BASIC-Statements angezeigt wird (Doppelpunkt nach der ersten Zahl).

- 1) Schalten Sie vom PkOgramm- in den RUN-Modus um.

Sie keinerlei Reaktion auf Tastenbetätigung erhalten.

- 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 PC 1350 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 Variablenamen gefolgt von  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 aufzufspüren - aber es ist manchmal die einzige.

Unabhängig davon, wie vorsichtig Sie auch programmieren, werden Sie ein Programm schreiben, welches nicht das macht, was Sie von dem Programm erwarten. Dann geht es darum, den Fehler zu finden. SHARPs Entwickler haben deshalb eine spezielle Programmablauf-Routine, genannt "Trace-Modus", vorbereitet. Im Trace-Modus stoppt der PC 1350 nach der Ausführung jeder einzelnen Programmzeile und zeigt die entsprechende Zeile nunmehr an. Das gibt Ihnen die Möglichkeit, die Ausführung des Programms in einzelnen Schritten nachzuprüfen. Hierbei haben Sie die Möglichkeit, sich nach jedem Programmschritt die Variablenwerte anzeigen zu lassen. Der Trace-Modus wird eingeschaltet durch Eingabe von "TRON". TRON kann als Kommando oder auch innerhalb eines Programms als Befehl verwendet werden.

wird TRON innerhalb eines Programms verwendet, so wird der Trace-Modus erst beim Erreichen der entsprechenden Zeilen eingeschaltet.

Ausschalten kann man den Trace-Modus mittels "TROFF". TROFF kann ebenso wie TRON als Kommando oder als Befehl innerhalb eines Programms verwendet werden. Den Trace-Modus kann man außerdem durch die folgende Tastenbetätigung ausschalten:

SHIFT  CA

Um ein Beispiel für die Benutzung des Trace-Modus zu haben, geben Sie bitte das folgende Programm ein, welches die Länge der Hypothense in einem rechtwinkligen Dreieck bei gegebener Länge der beiden anderen Seiten bestimmt:

```
10 INPUT A,B
20 A=A*A:B=B*B
30 H= (A+B)
40 PRINT "HYPOTHENUSE=" ;H
```

Schalten Sie um in den RUN-Modus, geben Sie das TRON-Kommando und starten Sie das Programm, in dem Sie das RUN-Kommando eingeben.

Beachten Sie, daß der INPUT-Befehl wie gewohnt abgearbeitet wird. Es erscheinen nacheinander die Fragezeichen für die Variablen, die eingegeben werden sollen. Nachdem Sie die zwei Werte eingegeben haben, erscheint die Zeilennummer des INPUT-Statements:

```
| 10: |
```

Drücken Sie jetzt die -Taste und halten Sie sie fest:

```
| 10: INPUT A,B |
```

\* Im Trace-Modus wird das errechnete Ergebnis an der Stelle, die durch einen CURSOR-Befehl spezifiziert wurde, ausgegeben, und die Zeilennummer wird in der darauffolgenden Zeile angezeigt.

\* Ist die Anzeige-Startposition mittels Cursor-Befehlen vorgegeben und werden nun manuell im Trace-Modus Variablen angezeigt oder Rechnungen ausgeführt, so wird die Anzeige-Startposition gelöscht.

Um das Programm fortzusetzen, drücken Sie die -Taste. Daraufhin wird die nächste Programmzeile ausgeführt und ihre Zeilennummer angezeigt.

Um die Programmzeile angezeigt zu bekommen, können Sie dann die -Taste betätigen. Die einzelnen Variablenwerte werden angezeigt, indem Sie die Variablennamen eingeben und die -Taste betätigen.

Beispiel:

EINGABE

ANZEIGE

A

```
| 4. |
```

(Sofern Sie 4 beim INPUT-Befehl für A eingegeben haben.)

Es ist notwendig, im Trace-Modus zur Abarbeitung jeder einzelnen Programmzeile die -Taste zu drücken. Betätigen Sie die -Taste, um einen solchen Programmablauf zu unterbrechen. Zur Programmfortführung geben Sie das CONT-Kommando.

Beispiel:

(Wir benutzen das Hypothenuse-Programm.)

EINGABE	ANZEIGE
	>
T R O N	TRON_
ENTER	>
R U N	RUN_
ENTER	?
3	3_
ENTER	?
4	4_
ENTER	10:
A	10:INPUT A,B
↓	20:
A	20:A=A*A : B=B*B
A	A_
ENTER	9.
B	B_
ENTER	16.
↓	30:
H	H_
ENTER	5
↓	HYPOTHENUSE=5.
A	40:PRINT "HYPOTHENUSE=" ;
↓	H
↓	40:
↓	>

## KAPITEL 11

## DIE INSTANDHALTUNG DES PC 1350

Um reibungsloses Funktionieren Ihres PC 1350 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 warmem 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!
- Wenn der Computer starker statischer Aufladung oder auch starkem Lärm 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.

## PROGRAMMBEISPIELE

Sie haben jetzt eine Reihe verschiedener Programmkommandos kennengelernt. Es ist nun nötig, selbst Programme mit Hilfe dieses Handbuchs zu erstellen. Nur so werden Sie Ihre Programmierfähigkeiten so weit verbessern, daß Sie sich vom Handbuch frei machen. Genau wie das Führen eines PKW oder das Tennisspielen werden Sie Ihre Fähigkeiten nur in der Praxis verbessern, d. h. indem Sie so viele Programme wie möglich erstellen. Genau so wichtig ist es aber, sich mit Programmen zu beschäftigen, die andere geschrieben haben. Vieles kann man erlernen, indem man versucht, ein fremdes Programm zu verstehen. Deshalb haben wir Ihnen hier eine Auswahl von BASIC-Programmen zusammengestellt.

(Für Schäden oder Verluste, die durch Programme dieses Handbuchs entstehen, übernehmen die SHARP-Corporation oder angegliederte Firmen keine Haftung.)

## INHALT

Programmname	Seite
* Mondlandung	P-2
* Berechnung der Anzahl der Tage zwischen zwei Daten	P-5
* ABC-Analyse	P-9
* Matrix-Berechnung	P-14
* Modifizierter beweglicher Mittelwert	P-19
* Einzelaufstellung	P-23
* F-Test	P-28

## MONDLANDUNG

Überblick: Treibstoff auf Reserve! NOTLANDUNG!

Wie landet man ein Raumschiff mit einem minimalen Rest an Treibstoff? Das Raumschiff befindet sich im freien Fall. Die Triebwerke müssen gezündet werden, um den Fall zu bremsen. Wird das Triebwerk zu früh oder zu lange gezündet, wird zu viel Treibstoff verbraucht und das Raumschiff wird zu glitzerndem Mondstaub. Wird das Triebwerk zu spät gezündet, donnern Sie in den Mondboden.

Ihre Aufgabe ist es also, die Nerven zu bewahren und das Raumschiff sicher und weich auf den Mond hinunterzubringen. Das Geheimnis ist der geschickte Einsatz des Triebwerks mit einem möglichst minimalen Treibstoffverbrauch.

Anleitung:

1. Das Programm wird mit **DEF A** gestartet.
2. Geben Sie mit **0** bis **9** die gewünschte Treibstoffmenge ein. Regulieren Sie vorsichtig den Treibstoffeinsatz, um das Raumschiff sicher zu landen. Haben Sie es geschafft, wird die verbliebene Treibstoffmenge angezeigt. Haben Sie Bruch gebaut, erscheint "Das war's!". Das Programm wird erst fortgesetzt, wenn Sie eingeben, ob Sie weiterspielen wollen oder nicht.

Aufbau:

Die Anziehung wird auf  $5 \text{ m/sec}^2$  festgelegt. Werden in einer Sekunde 5 Einheiten Treibstoff verbrannt, ist die Anziehung neutralisiert.

H	= Höhe
H0	= Anfangshöhe
V	= Geschwindigkeit
V0	= Anfangsgeschwindigkeit
a	= Anziehung
F	= verbrauchter Treibstoff
F0	= Anfangstreibstoff
t	= Zeit

$$H = H0 + V0*t + 1/2 * a*t*t$$

$$V = V0 + a*t$$

$$V*V = V0*V0 + 2*a*H$$

$$H0 = 500, V0 = -50,$$

$$F0 = 200$$



1.  DEF  A Programmstart

***MONDLANDUNG***			
Höhe	Geschw.	TR	TV
500	-50	200	0

2.  9 Treibstoffeinsatz

Höhe	Geschw.	TR	TV
452	-46	191	9

3.  6 Treibstoffeinsatz

Höhe	Geschw.	TR	TV
420	-50	185	6

usw.

4. Bei erfolgreicher Landung:

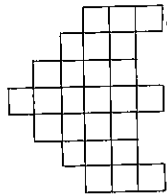
Höhe	Geschw.	TR	TV
0	0	80	0

\*\* Sauber gelandet \*\*  
Treibstoffrest: 80  
Noch einmal? (J/N)

5. Bei Bruchlandung:

Höhe	Geschw.	TR	TV
12	-70	100	6
*** das war's ***			
Noch einmal? (J/N)			

Punktform der Graphik:



## PROGRAMMLISTING

```

10 "A": WAIT 0: CLEAR : USING : CLS
20 S=-50:A=0:D$="":V=30
30 CURSOR 0,1: BEEP 3: CLS: PRINT "***MONDLANDUNG***"
40 DATA "Zeit=",5
50 DATA "Treibstoff=",200
60 DATA "Hoehe=",500
70 RESTORE : READ B$,W,B$,F,B$,H
80 WAIT w
90 FOR P=1 TO 2: GCURSOR (135,P*7): GPRINT
"FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF": NEXT P
100 CURSOR 0,2:PRINT "Hoehe Gesch. Trst. TV"
110 GOSUB 300
120 L=V
130 IF F<=0 LET C=0: BEEP 1: GOTO 180
140 BEEP 1: D$= INKEY$
150 IF D$="" LET C=A: GOTO 180
160 C= VAL D$
170 A=C
180 IF C>F LET C=F
190 F=F-C:X=C-5:H=H+S+X/2:S=S+X
200 IF H>0 LET V=(500-H)/5+30: GOTO 110
210 IF ABS H<5 OR ABS S<5 LET H=0:S=0:L=V:V=129: GOSUB 300
215 BEEP 5: CLS :PRINT "***Saubere gelandet***":GOTO 250
220 GCURSOR (L,11): GPRINT "000000000000"
230 GCURSOR (129,11): GPRINT "15A006A00A2108"
240 BEEP 3: CLS : CURSOR 0,1: PRINT "***
war";CHR$ 39;"s ***": GOTO 260
250 PRINT USING "###";"Treibstoffrest: ";F
260 CURSOR 0,3: PRINT "Noch einmal? (J/N)"
270 Z$= INKEY$: IF Z$="J" THEN 10
280 IF Z$="N" END
290 GOTO 270
300 CURSOR 0,3: PRINT USING "###";H; USING
"###";S;F USING "###";C
310 GCURSOR (L,11): GPRINT "000000000000"
320 GCURSOR (V,11): GPRINT "081C3E7F7F49"
330 RETURN

```

## SPEICHERINHALTE

A	
B\$	
C	Eingesetzter Treibstoff
D\$	Eingesetzter Treibstoff
F	Verbliebener Treibstoff
H	Höhe
L	
P	Schleifenvariable
S	Geschwindigkeit
V	
W	Wartezeit
X	
Z\$	

## BERECHNUNG DER ANZAHL DER TAGE ZWISCHEN ZWEI DATEN

Überblick:

Dieses Programm kann zwei verschiedene Berechnungen ausführen. Es berechnet Ihnen die Zahl der Tage, die zwischen zwei bekannten Daten liegen, oder es ermittelt Ihnen das Datum zu einem Tag nach dem soundsovielten. Alles, was Sie eingeben, sind entweder das Startdatum und das Zieldatum, oder das Startdatum und die Zahl der Tage zum Zieldatum. Geben Sie doch einmal Ihren Geburtstag und das heutige Datum ein, oder lassen Sie das Datum von heute in 10.000 Tagen ermitteln.

Anweisungen:

DEF  A Berechnung der Anzahl der Tage.

1. Geben Sie das Startjahr, den Monat und den Tag in den Computer ein. Soll eine Eingabe korrigiert werden, geben Sie dann ein  N, ist die Eingabe korrekt, geben Sie ein  Y ein.
2. Geben Sie dann das Zieljahr, den Monat und den Tag ein. Auch hier können Sie wieder korrigieren, wenn nötig. Wenn Sie ein "Y" eingeben, wird die Anzahl der Tage zwischen den beiden gegebenen Daten ausgegeben.
3. Soll das Programm wiederholt werden, drücken Sie die  ENTER - Taste. Um das Programm zu beenden, muß bei der Startdatum-Eingabe ohne Daten die  ENTER -Taste gedrückt werden.

DEF  B Ermittlung des Datums nach x Tagen

Der Programmablauf ist der gleiche wie mit  DEF  A. Nur, daß Sie statt eines Zieldatums die Zahl der Tage eingeben müssen und Ihnen das zugehörige Datum ermittelt wird.

## Beispiele:

1. Berechnung der Anzahl der Tage

Startdatum: 7. Februar 1961  
Zieldatum: 9. Mai 1984

2. Ermittlung des Datums

Startdatum: 7. Februar 1961  
Zahl der Tage: 8.492

## Tastenfolge zu 1.:

- |    |   |  |
|----|---|--|
| 1) | <input type="text" value="DEF"/> <input type="text" value="A"/>   | Programmstart  |
| 2) | 1961 <input type="text" value="ENTER"/>                           | Eingabe des Startdatums  |
|    | 2 <input type="text" value="ENTER"/>                              |  |
|    | 3 <input type="text" value="ENTER"/>                              |  |
| 3) | <input type="text" value="Y"/> <input type="text" value="ENTER"/> | Keine Korrekturen  |
| 4) | 1984 <input type="text" value="ENTER"/>                           | Eingabe des Zieldatums   |
|    | 5 <input type="text" value="ENTER"/>                              |  |
|    | 9 <input type="text" value="ENTER"/>                              |  |
| 5) | <input type="text" value="Y"/> <input type="text" value="ENTER"/> | Keine Korrekturen  |
| 6) | Ausgabe 8492  | Der Rechner gibt die Zahl der Tage<br>zwischen den beiden Daten aus. |
| 7) | <input type="text" value="ENTER"/>                                | Neustart des Programms   |
| 8) | <input type="text" value="ENTER"/>                                | Beenden des Programms  |

## Tastenfolge zu 2.:

- |     |   |   |
|-----|---|---|
| 1)  | <input type="text" value="DEF"/> <input type="text" value="B"/>   | Programmstart                           |
| 2)  | 1960 <input type="text" value="ENTER"/>                           | Eingabe des Startdatums                 |
|     | 2 <input type="text" value="ENTER"/>                              |   |
|     | 7 <input type="text" value="ENTER"/>                              |   |
| 3)  | <input type="text" value="N"/> <input type="text" value="ENTER"/> | Sie haben einen Eingabefehler gefunden. |
| 4)  | 1961 <input type="text" value="ENTER"/>                           | Korrektur des Startdatums               |
|     | 2 <input type="text" value="ENTER"/>                              |   |
|     | 7 <input type="text" value="ENTER"/>                              |   |
| 5)  | <input type="text" value="Y"/> <input type="text" value="ENTER"/> | Keine Korrekturen                       |
| 6)  | 8492 <input type="text" value="ENTER"/>                           | Eingabe der Zahl der Tage               |
| 7)  | <input type="text" value="Y"/> <input type="text" value="ENTER"/> | Keine Korrekturen                       |
| 8)  | AUSGABE Y=1984 M=5 D=9  | Ausgabe des ermittelten Datums          |
| 9)  | <input type="text" value="ENTER"/>                                | Neustart des Programms                  |
| 10) | <input type="text" value="ENTER"/>                                | Beenden des Programms                   |

## PROGRAMMBEISPIELE

## PROGRAMMLISTING

```

10 REM *****
20 REM NO. OF DAYS CALC.
30 REM *****
40 A : CLEAR : CLS : WAIT 0: USING :A=0: GOTO 50
45 B : CLEAR : CLS : WAIT 0: USING :A=1
50 PRINT "*** NO. OF DAYS CALC: ***"
60 BEEP 1
80 CURSOR 0,1: PRINT "START Y=";" M=";" D= "
85 IF A=1 CURSOR 0,2: PRINT "DAYS = "": GOTO 95
90 CURSOR 0,2: PRINT "TARGET Y=";" M=";" D= "
95 CURSOR 0,3: PRINT
97 IF C=1 THEN 180
100 CURSOR 9,1: GOTO 105
105 INPUT R: GOTO 120
110 CLS : END
120 CURSOR 16,1: GOTO 125
125 INPUT S
130 CURSOR 21,1: GOTO 135
135 INPUT T
140 CURSOR 0,3: PRINT " "
150 CURSOR 13,3: GOTO 155
155 INPUT "OK(Y/N)?" ;F$
160 IF F$="N" THEN 80
165 IF F$<>"Y" GOTO 150
170 CURSOR 0,3: PRINT " "
180 CURSOR 9,2: GOTO 185
185 INPUT U
189 IF A=1 LET X=U: GOTO 210
190 CURSOR 16,2: GOTO 195
195 INPUT V
200 CURSOR 21,2: GOTO 205
205 INPUT W
210 CURSOR 13,3: GOTO 215
215 INPUT "OK(Y/N)?" ;F$
220 IF F$="N" LET C=1: GOTO 85
225 IF F$<>"Y" THEN 210
230 CURSOR 0,3: PRINT ""PLEASE WAIT ! ! "
235 IF A=1 THEN 400
240 H=R:G=S:I=T: GOSUB 300:J=I:H=U:G=V:I=W: GOSUB 300:X=I-J
250 BEEP 1: CURSOR 0,3: PRINT " "
260 CURSOR 0,3: PRINT "NO. OF DAYS =>"; USING "#####";X
270 CURSOR 20,3: GOTO 275
275 INPUT " ";K
280 USING :C=0: GOTO 80
300 IF G-3>=0 LET G=G+1: GOTO 320
310 G=G+13:H=H-1
320 I= INT (365.25*H)+ INT (30.6*G)+I:I=I- INT (H/100)+ INT (H/400)-428:
RETURN
400 H=R:G=S:I=T: GOSUB 600:J=Q+X:G=1:I=0
410 H=H+1: GOSUB 600
420 IF J>Q THEN 410
430 H=H-1
440 G=G+1: GOSUB 600
450 IF J>Q THEN 440

```

PROGRAMMBEISPIELE

```

460 G=G-1: GOSUB 600:I=J-Q:H=H-B: BEEP 1: CURSOR 0,3: PRINT "
"
470 CURSOR 0,3: PRINT "TARGET Y=";" M=";" D= "
475 CURSOR 9,3: PRINT H: CURSOR 16,3: PRINT G: CURSOR 21,3: PRINT I
480 CURSOR 21,2: GOTO 485
485 INPUT " ";K
490 GOTO 80
600 IF G-3>=0 LET O=G+1:P=H: GOTO 620
610 O=G+13:P=H-1
620 Q= INT (365.25*P)+ INT (30.6*O)+I:Q=Q- INT (P/100)+ INT (P/40)
-428: RETURN
    
```

SPEICHERINHALTE

A		P	
B		Q	
C		R	Startjahr
F\$		S	Startmonat
G		S	Starttag
H		U	Zieljahr
I		V	Zielmonat
J		W	Zieltag
K		X	Zahl der Tage
O			

## ABC-ANALYSE

Konfiguration: CE-126P Drucker/Cassetten-Interface wird benötigt.

Das Programm ermittelt die Produktrangfolge. Für einen Verkaufsleiter ist es wichtig zu wissen, welche Position ein Produkt in der Tabelle der Verkaufszahlen einnimmt. Mit dem Programm ABC-Analyse können Sie den Computer ermitteln lassen, wie die Rangfolge bezüglich der Verkaufszahlen aussieht. Sie geben einfach die Verkaufszahlen und die Produktnummer ein. Der Computer druckt Ihnen dann die Verkaufsraten und die Gesamtsumme aus und ordnet die Produkte nach der ABC-Analyse.

Anweisungen:

1. Das Programm wird mit dem Kommando RUN gestartet. Geben Sie entsprechend der Bildschirmanzeige die Produktnummer und die Verkaufszahlen ein. Wenn Sie statt der Produktnummer nur die  [ENTER] -Taste drücken, kommen Sie zum folgenden Schritt:
2. Entscheiden Sie, ob Sie die nächsten Daten übernehmen wollen. Geben Sie  [Y] ein, wenn sie übernommen werden sollen. Korrigieren Sie die Daten, wenn nötig, indem Sie die richtigen Werte eingeben. Geben Sie ein  [N] ein, wenn keine Berichtigung nötig ist.
3. Ausgabe der Verkaufsraten, der Gesamtsumme und der ABC-Analyse. Soll die Ausgabe über den Drucker erfolgen, geben Sie ein  [P] ein. Ein  [L] gibt die Ausgabe auf den LCD-Bildschirm. Mit  [U] und  [D] (Up = hoch; Down = runter) können Sie die Ausgabe verschieben.  [E] wie Ende beendet das Programm.
4. Wenn Sie das Ergebnis ausdrucken lassen oder nach der Ausgabe auf dem Bildschirm ein  [Q] eingeben, kommen Sie wieder zu 3.

Achtung: Produktnummern können bis zu 7 Zeichen lang sein, Verkaufszahlen bis zu 6 Ziffern, maximal 26 Posten können eingegeben werden.

## Beispiele:

Folgende Daten dienen als Basis für die Berechnung der Verkaufsraten und der Gesamtsumme sowie der ABC-Analyse.

Produktnummer	Verkaufszahl
TR-300	890
BT-650	170
GZ-90	230
RS-15	650
JO-230	560
PQ-180	1080
YC-30	210

## GEDRUCKTE AUSGABE

\*\* ABC ANALYSIS \*\*

*PROD. NO.*	(RANK)		
SALES %	SUM %		
-----			
* PQ-180 *	( A )		
108000 28.4	108000	28.4	
-----			
* TR-300 *	( A )		
89000 23.4	197000	51.9	
-----			
* RS-15 *	( B )		
65000 17.1	262000	69.1	
-----			
* JO-230 *	( B )		
56000 14.7	318000	83.9	
-----			
* GZ-90 *	( B )		
23000 6.0	341000	89.9	
-----			
* YC-30 *	( B )		
21000 5.5	362000	95.5	
-----			
* BT-650 *	( C )		
17000 4.4	379000	100	
-----			

## Tastenfolge:

- |           |                                    |                                     |
|-----------|------------------------------------|-------------------------------------|
| 1) RUN    | <input type="text" value="ENTER"/> | Programmstart                       |
| 2) TR-300 | <input type="text" value="ENTER"/> | Eingabe der 1. Produktnummer        |
| 3) 880    | <input type="text" value="ENTER"/> | Eingabe der 1. Verkaufszahl         |
| :         |                                    |                                     |
| :         |                                    |                                     |
| :         |                                    |                                     |
| 4) YC-30  | <input type="text" value="ENTER"/> | Eingabe der 7. Produktnummer        |
| 5) 210    | <input type="text" value="ENTER"/> | Eingabe der 7. Verkaufszahl         |
| 6)        | <input type="text" value="ENTER"/> | Soll Datenkontrolle gewählt werden? |
| 7) Y      | <input type="text" value="ENTER"/> | Wahl der Datenkontrolle             |
| 8)        | <input type="text" value="ENTER"/> |                                     |
| 9) 090    | <input type="text" value="ENTER"/> | Korrektur der Daten                 |
| :         |                                    |                                     |
| :         |                                    |                                     |
| :         |                                    |                                     |



10)	<input type="text" value="ENTER"/>	Datum korrekt
11)	<input type="text" value="ENTER"/>	Erneute Prüfung der Daten?
12)	N <input type="text" value="ENTER"/>	Keine Prüfung
13)	L <input type="text" value="ENTER"/>	Ausgabe des Ergebnisses auf dem Bildschirm
14)	D	Bildschirminhalt nach unten rollen
	.	
	.	
	.	
	.	
	.	
	.	
15)	D	Bildschirminhalt nach unten rollen
16)	Q	Zurück zum Ausgabemenü
17)	P <input type="text" value="ENTER"/>	Ausgabe über Drucker
18)	E <input type="text" value="ENTER"/>	Programm beenden

PROGRAMMLISTING

```

40 CLEAR : CLS : WAIT 0:A=26:B=14: DIM B$(A)*B,C(A),D(A),K$(A)*1,Z$(0)*6
50 FOR I=1 TO A: CURSOR : IF I>2 GOSUB 700: PRINT " "
60 CURSOR 0,(I>1)*2: PRINT "("; STR$ I;") PRODUCT NO. SALES": PRINT " "
70 BEEP 1: CURSOR 2+ LEN STR$ I,(I=1)+(I>1)*3: GOTO 75
75 INPUT B$(I)
80 IF B$(I)="" LET C=I-1:I=A: GOTO 130
90 B$(I)= LEFT$( B$(I),7)
95 CURSOR 2+ LEN STR$ I,(I=1)+(I>1)*3: PRINT B$(I);" "
100 CURSOR 14+ LEN STR$ I,(I=1)+(I>1)*3: BEEP 1:GOTO 105
105 INPUT Z$(0)
107 CURSOR 14+ LEN STR$ I,(I=1)+(I>1)*3: PRINT Z$(0);" "
110 C(I)= VAL Z$(0): IF C(I)<=0 BEEP 1: GOTO 100
120 E=E+C(I):C=I:Z$(0)=""
130 NEXT I: IF C<2 BEEP 2: GOTO 40
140 BEEP 1: CLS : INPUT "CONFIRM OR AMEND(Y/N)";Z$
150 IF Z$="N" GOSUB 520: GOTO 240
160 IF Z$<>"Y" THEN 140
170 CLS : FOR I=1 TO C
180 CURSOR : IF I>2 GOSUB 700: PRINT " "
190 CURSOR 0,(I>1)*2: PRINT "("; STR$ I;");B$(I);" -> ": CURSOR 6+ LEN
STR$ I + LEN B$(I),(I>1)*2: GOTO 200
200 INPUT B$(I):B$(I)= LEFT$( B$(I),7)
210 CURSOR 2+ LEN STR$ I,(I=1)+(I>1)*3: PRINT STR$ C(I);" ->": CURSOR
LEN STR$ C(I)+7,(I=1)+(I>1)*3
215 Z$(0)= STR$ C(I): GOTO 220
220 INPUT Z$(0): IF VAL Z$(0)<=0 BEEP 1: GOTO 210
230 E=E-C(I):C(I)= VAL Z$(0):E=E+C(I): NEXT I: CLS : GOTO 140
240 CLS : H=0:V=0: PRINT "LCD or PRINTER or END"
250 BEEP 1: CURSOR 7,1: GOTO 255
255 INPUT "(L/P/E)?" ;Z$
260 IF Z$="E" BEEP 3: CLS : END
270 IF Z$="P" THEN 430
280 IF Z$<>"L" THEN 240
290 CLS : CURSOR 0,0: PRINT "PR. NO. RATE(%)": CURSOR 16,0: PRINT
" SUM(%)R":Y=1
300 FOR I=1 TO 3-(C=2): CURSOR 0,I: GOSUB 700:V=D(Y-1)
310 CURSOR 0,1: PRINT B$(Y): CURSOR 8,I:O$= STR$ ( INT (D(Y)-V))
315 IF LEN O$=1 LET O$=""+O$
317 PRINT O$
320 CURSOR (64+D(Y)/2)/6-.9,I: PRINT USING "###.###":D(Y): CURSOR 23,I:
PRINT K$(Y)
330 LINE (66,I*8+1)-(66+(D(Y)-V)/2,I*8+5),BF
340 LINE (66+(D(Y)-V)/2,I*8+1)-(66+D(Y)/2,I*8+5),B
350 Y=Y+1: NEXT I
360 LINE (2,7)-(149,7): LINE (2,31)-(149,31): LINE (2,0)-(2,31)
370 LINE (48,0)-(48,31): LINE (143,0)-(143,31): LINE (149,0)-(149,31)
380 IF INKEY$="" GOTO 380
390 IF INKEY$="Q" GOTO 240
400 IF INKEY$="D" AND Y<C+1 BEEP 1:Y=Y-2: GOTO 300
410 IF INKEY$ <>"U" OR Y<5 BEEP2: GOTO 380
420 Y=Y-4: BEEP 1: GOTO 300
430 LPRINT " ** ABC ANALYSIS **": LPRINT
440 LPRINT "*PROD. NO.* (RANK)"
450 LPRINT " SALES % SUM %"
460 FOR I=1 TO C: GOSUB 690:H=H+C(I)

```

```

470 LPRINT "*" ; USING "#####";B$(I); USING ";" * (";K$(I);"
)
480 LPRINT USING "#####": STR$ C(I); USING "###.#";D(I)-V: USING
"#####";H;
490 IF D(I)<>100 USING "###.#": GOTO 510
500 USING "#####"
510 LPRINT D(I):V=D(I): NEXT I: GOSUB &=: LPRINT : LPRINT : GOTO 240
520 L= INT (C/2+1):R=C
530 IF L>1 LET L=L-1:X=C(L):B$(0)=B$(L): GOTO 560
540 X=C(R):B$(0)=B$(R):C(R)=C(1):B$(R)=B$(1):R=R-1
550 IF R<=1 LET C(1)=X:B$(1)=B$(0): GOTO 630
560 J=L
570 I=J:J=2*J
580 IF J>R THEN 620
590 IF J<R THEN IF C(J)>C(J+1) LET J=J+1
600 IF X<=C(J) THEN 620
610 C(I)=C(J):B$(I)=B$(J): GOTO 570
620 C(I)=X:B$(I)=B$(0): GOTO 530
630 FOR I=1 TO C
640 H=H+C(I):D(I)=H/E*100
650 IF D(I)<=66 OR I=1 LET K$(I)="A": GOTO 680
660 IF D(I)<=99 LET K$(I)="B": GOTO 680
670 K$(I)="C"
680 NEXT I: RETURN
690 USING : LPRINT "-----": RETURN
700 PRINT " " " : RETURN
    
```

SPEICHERINHALT

A	maximale Zahl der Datensätze	V	
B	Lage der Produktnummer	X	
C	Zahl der Datensätze	Y	
E	Verkaufszahlensumme	Z\$	
H		C(A)	Verkaufszahlen
I	Schleifenvariable	D(A)	
J	Schleifenvariable	B\$(A)	Kundenname
L		K\$(A)	
O\$		Z\$(0)	

## MATRIXBERECHNUNG

Konfiguration: CE-126P Drucker/Cassetten-Interface wird benötigt.

Überblick:

Mit diesem Programm können Sie einfache Fehler vermeiden. Obwohl die Berechnungsmethode selber ziemlich leicht verständlich ist, geht viel Zeit dadurch verloren, die Berechnung auf einfache Fehler zu überprüfen. Genau hierbei soll Ihnen das Programm helfen. Wir stellen Ihnen hier ein Programm vor, das die Problemwerte einer Matrix herausucht. Die Korrektur und der Ausdruck der eingegebenen Werte sind möglich.

Anweisungen:

1.  DEF  A startet das Programm.
  - \* Zuerst wird die Dimensionierung der Matrix eingegeben, dann die Daten der Matrix.
  - \* Sind alle Daten eingegeben, können Sie wählen, ob sie ausgedruckt werden sollen oder nicht.
2. Danach können Sie jedes eingegebene Datum überprüfen und wenn nötig, korrigieren.
  - \* Soll korrigiert werden, geben Sie  Y  ENTER ein, dann die Zeile und die Spalte des zu korrigierenden Wertes. Geben Sie den richtigen Wert ein.
  - \* Ist keine Korrektur nötig, geben Sie  N  ENTER ein.
3. Ausgabe der Berechnung und Endkontrolle
  - \* Nach der Überprüfung der Werte und Korrektur aller Fehleingaben können Sie erneut wählen, ob die Daten ausgedruckt werden sollen oder nicht. Danach ist eine weitere Kontrolle der Daten und deren Korrektur möglich. Geben Sie  N  ENTER ein, wenn Sie keine Kontrolle wünschen.

Inhalt:

Das Programm konvertiert die Matrix nach der "Sweeping-Out"-Methode in eine Dreiecksmatrix.

Die Matrix sei

$$\begin{aligned}
 & \{a_{ij}\} \quad (i, j = 1 \sim n) \\
 P &= a_{mm} \quad (m = 2 \sim n) \\
 q &= a_{im}/P \quad (i = 1 \sim m-1) \\
 a_{ij} &= a_{ij} - q \cdot a_{mj} \quad (j = 1 \sim m)
 \end{aligned}$$

Die Berechnung ergibt:

$$a_{ij} = 0 \text{ für } i < j$$

Daraus ergibt sich:

$$\det = a_{11} * a_{22} * a_{33} \dots a_{nn}$$

Beispiele:

Suche nach der Determinanten folgender Matrix:

$$\begin{vmatrix} 4 & 7 & 1 & 8 \\ 5 & 3 & 9 & 3 \\ 2 & 5 & 3 & 5 \\ 1 & 4 & 7 & 8 \end{vmatrix}$$

Druckbeispiel:

$A(1,1)=4.$   
 $A(1,2)=7.$   
 $A(1,3)=1.$   
 $A(1,4)=8.$   
 $A(2,1)=5.$   
 $A(2,2)=3.$   
 $A(2,13)=9.$   
 $A(2,4)=3.$   
 $A(3,1)=2.$   
 $A(3,2)=5.$   
 $A(3,3)=3.$   
 $A(3,4)=5.$   
 $A(4,1)=1.$   
 $A(4,2)=4.$   
 $A(4,3)=7.$   
 $A(4,4)=8.$

$\det = -525.$

## Tastenfolge:

1.  DEF  A Programmstart
2. 4  ENTER Dimensionierung der Matrix
- 3) 4  ENTER Eingabe des 1. Elements A(1,1)
- 4) 6  ENTER Eingabe des 2. Elements A(1,2)
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 5) 8  ENTER Eingabe des letzten Elements A(4,4)
- 6)  N  ENTER Kein Ausdruck der Werte
- 7)  Y  ENTER Überprüfung der Daten und Korrektur
- 8) 1  ENTER Eingabe der Spalte
- 9) 2  ENTER Eingabe der Zeile
- 10) 7  ENTER Eingabe des korrekten Wertes
- 11)  N  ENTER Keine weitere Korrektur
- 12)  Y  ENTER Datenliste ausdrucken
- 13)  N  ENTER Keine Korrekturen, Beendigung des Programms

PROGRAMMLISTING

```

10 REM *****
20 REM * MATRIX CALC. *
30 REM *****
40 "A": CLEAR : CLS : WAIT 0
50 INPUT "DEGREES=";N
60 N=N-1
70 DIM A(N,N)
80 GOSUB 230
90 F=0
100 INPUT "PRINT OUT DATA (Y/N)";O$
110 IF O$="N" THEN 140
120 IF O$ <>"Y" THEN 90
130 GOSUB 350
140 INPUT "AMEND DATA (Y/N)";O$
150 IF (O$="N")*(F=0) THEN 200
160 IF (O$="N")*(F=1) THEN 90
170 IF O$<>"Y" THEN CURSOR 0,3: GOTO 140
180 GOSUB 430
190 GOTO 140
200 GOSUB 580
210 CLS
220 END
230 L=1
240 FOR I=0 TO N
250 FOR J=0 TO N
260 A$="A("+ STR$(I+1)+"," + STR$(J+1)+")="
270 PRINT A$;
280 CURSOR 7,L: GOTO 290
290 INPUT A(I,J)
300 L=L+1: IF L=4 THEN LET L=3
310 CURSOR
320 NEXT J
330 NEXT I
340 RETURN
350 FOR I=0 TO N
360 FOR J=0 TO N
370 A$="A("+ STR$(I+1)+"," +STR$(J+1)+")="
380 LPRINT A$;A(I,J)
390 NEXT J
400 NEXT I
410 LPRINT : LPRINT
420 RETURN
430 CLS
440 PRINT "X=      Y="
450 CURSOR 2,0: GOTO 455
455 INPUT X
460 IF (X<1)+(X>N+1) THEN 450
470 CURSOR 10,0: GOTO 475
475 INPUT Y
480 IF (Y<1)+(Y>N+1) THEN 470
490 CURSOR
500 A$="A(" + STR$(X)+"," + STR$(Y)+")="
510 PRINT A$;A(X-1,Y-1)
520 PRINT A$
530 CURSOR 7,2: GOTO 540

```

```

540 INPUT A(X-1,Y-1)
550 CURSOR
560 F=1
570 RETURN
580 For M=N TO 1 STEP -1
590 P=A(M,M)
600 IF P=0 THEN 760
610 For I=0 TO M-1
620 Q=A(I,M)/P
630 FOR J=0 TO M
640 A(I,J)=A(I,J)-Q*A(M,J)
650 NEXT J
660 NEXT I
670 NEXT M
680 D=A(0,0)
690 FOR I=1 TO N
700 D=D*A(I,I)
710 NEXT I
720 BEEP 1
730 LPRINT "det= ";D
740 LPRINT : LPRINT
750 RETURN
760 BEEP 2: CLS
770 PRINT "CALCULATE IMPOSSIBLE !!"
780 END

```

## Speicherinhalte

A\$	Anzeige	N	Dimension
D	Matrix	O\$	Flag (Marke)
F	Flag	P	
I	Zähler	Q	
J	Zähler	X	Wechsel eines Matrixelements
L		Y	Wechsel eines Matrixelements
M	Zähler	A(N,N)	Elemente der Matrix



## MODIFIZIERTER BEWEGLICHER MITTELWERT

Überblick:

Selbst schwer erkennbare Veränderungen können leicht erfaßt werden:

Der modifizierte bewegliche Mittelwert wird überall dort eingesetzt, wo aus einer festen Datenfolge der Mittelwert an jedem Punkt ermittelt werden soll oder muß, um Datentrends zu erkennen.

Konkret kann mit diesem Programm zum Beispiel die Entwicklung der Verkaufszahlen deutlich gemacht werden, der Einkaufspreise oder die Wechselkurse, Basis für die großen täglichen Bewegungen.

Anweisungen:

1. DEF A startet das Programm. Geben Sie dann die Anzahl der Werte und dann die einzelnen Werte ein. Wird ein Wert eingegeben, so wird dieser Wert und der jetzige Mittelwert auf dem Drucker ausgegeben.
2. Sind alle Daten eingegeben, können Sie das Programm beenden, indem Sie bei der Abfrage einer Eingabe einfach die ENTER - Taste drücken.

Inhalt:

Es gibt zwei verschiedene Versionen, je nachdem, ob die Zahl der Eingaben (n) gerade oder ungerade ist.

1. n ist eine ungerade Zahl:

$$X_1 = \sum_{i=1}^n X_i / n$$

$$X_2 = \sum_{i=1}^{n-1} X_i / n$$

⋮

2. n ist eine gerade Zahl:

$$X_1 = \left( \frac{X_1 + X_{n+1}}{2} + \sum_{i=1}^n X_i \right) / n$$

$$X_2 = \left( \frac{X_2 + X_{n+1}}{2} + \sum_{i=1}^{n-1} X_i \right) / n$$

⋮

Beispiel:

Geben Sie die folgenden Daten ein, um den modifizierten beweglichen Mittelwert zu ermitteln. Setzen Sie dazu die Zahl der Eingaben auf 4.

X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>	X <sub>6</sub>	X <sub>7</sub>	X <sub>8</sub>	X <sub>9</sub>	X <sub>10</sub>	X <sub>11</sub>	X <sub>12</sub>	X <sub>13</sub>
82	15	8	53	0	41	22	3	18	75	64	39	46

Druckerausgabe:

```
* X(1)=82.
* X(2)=15.
* X(3)=8.
* X(4)=53.
* X(5)=0.
Average=29.25
* X(6)=41.
Average=22.25
* X(7)=22.
Average=27.25
* X(8)=3.
Average=22.75
* X(9)=18.
Average=18.75
* X(10)=75.
Average=25.25.
* X(11)=64.
Average=34.75
* X(12)=39.
Average=44.5
* X(13)=46.
Average= 52.5
```

Tastenfolge:

- 1) DEF A                    Programmstart
- 2) 4 ENTER                Eingabe der Anzahl der Werte
- 3) 83 ENTER               Eingabe von X(1)
- :
- :
- :
- :
- :
- 4) 46 ENTER               Eingabe von X(13)
- 5) ENTER                   Programmende

PROGRAMMLISTING

```

10 REM *****
20 REM MODIFIED MOVING
25     AVERAGE
30 REM *****
40 "A": CLEAR : CLS : WAIT 0
50 N=1:E=0:C=1
60 INPUT "NUMBER OF ITEMS=";A
70 DIM X(A-1)
80 IF A<> INT (A/2)*2 THEN 220
90 FOR I=0 TO A-1
100 GOSUB 430
110 NEXT I
120 FOR I=0 TO A-1
130 PRINT "X("+ STR$ (N)+)"=";
140 CURSOR 5+ LEN ( STR$ (N)),C: GOTO 145
145 INPUT D: GOTO 160
150 LPRINT : LPRINT : CLS : END
160 E=E+D
170 LPRINT "* X("+ STR$(N)+)"=";D
180: LPRINT " AVERAGE=";(E-.5*(D+X(I)))/A
190 E=E-X(I):X(I)=D:N=N+1:C=C+1: IF C=4 LET C=3
200 NEXT I
210 GOTO 120
220 FOR I=0 TO A-2
230 GOSUB 430
240 NEXT I
250 B=A-1
260 PRINT "X("+ STR$ (N)+)"=";
270 CURSOR 5+ LEN (STR$ (N)),C: GOTO 280
280 INPUT X(B)
290 LPRINT "* X("+ STR$ (N)+)"=";X(B)
300 LPRINT " AVERAGE="; E/A
310 E=E+X(B):N=N+1:C=C+1: IF C=4 LET C=3
320 FOR I=0 TO B
330 PRINT "X("+ STR$ (N)+)"="
340 CURSOR 5+ LEN ( STR$ (N)),C: GOTO 350
350 INPUT D: GOTO 370
360 LPRINT : LPRINT : CLS : END
370 E=E-X(I)+D:X(I)=D
380 LPRINT "* X("+ STR$ (N)+)"=";D
390 LPRINT " AVERAGE="; E/A
400 N=N+1:C=C+1: IF C=4 LET C=3
410 NEXT I
420 GOTO 320
430 PRINT "X("+ STR$ (N)+)"=";
440 CURSOR 5+ LEN ( STR$ (N)),C: GOTO 450
450 INPUT D
460 LPRINT "* X("+ STR$ (N)+)"=";D
470 E=E+D:X(I)=D:N=N+1:C=C+1: IF C=4 THEN LET C=3
480 RETURN

```

## Speicherinhalt

A	Anzahl der Werte	E	Datensumme
B		I	Schleifenvariable
C		N	
D	Daten	$X(A-1)$	Daten

EINZELAUFSTELLUNG

Konfiguration: CE-126P Drucker/Cassetten-Interface wird benötigt.

Überblick:

Eine Hilfe für die Bruchanalyse.

Dieses Programm soll die Schwierigkeiten mit der Vielzahl von Ebenen und Abbildungen lösen, und die Ecksumme (S), den Freiraum (DF), die objektive Bewertung der Populationsdichte-Veränderung (V) sowie das Verhältnis der objektiven Bewertung der Populationsdichte-Veränderung (f) zu jedem Wert berechnen.

Anweisungen:

1.  DEF  A startet das Programm.
- \* Wählen Sie die Ausgabemethode;  1  ENTER für den Drucker und  2  ENTER für den Bildschirm.
- \* Geben Sie dann die Zahl der Ebenen und die Zahl der Abbildungen ein. Es folgt die Eingabe der Werte, die vom Rechner erwartet und auf dem Bildschirm abgefragt werden.
- \* Wurden alle Daten eingegeben, können Sie wählen, ob die Eingaben ausgedruckt werden sollen oder nicht.
2. Entscheiden Sie dann, ob Sie Daten korrigieren wollen.
- \* Für die Korrektur geben Sie  Y  ENTER ein. Danach geben Sie die Zeile und die Spalte des zu korrigierenden Werts ein. Korrigieren Sie den Wert.
- \* Ist keine Korrektur nötig, geben Sie  N  ENTER ein.
3. Ausgabe der Berechnungen.
- \* Ist die Datenkorrektur abgeschlossen, können Sie erneut wählen, ob die Ausgabe auf dem Drucker oder dem Bildschirm erfolgen soll. Sie können wieder Korrekturen vornehmen, wenn Sie vorher  N  ENTER eingegeben haben.

Inhalt:

Zahl der Ebenen: a  
Zahl der Abbildungen: n

Werte:  $x_{ij}$  ( $i=1\sim a, j=1\sim n$ ) Zahl der Werte an

1.	2.	3.	4.	5.
$(X) = x^i / an$	$S_A = [A] - (X)$	$\phi_A = a - 1$	$(V) = (S) / (\phi)$	$F = V_A / V_E$
$(A) = x^i / n$	$S_T = [AS] - (X)$	$\phi_X = a_n - a$		
$(AS) = \sum x^i_j$	$S_E = [AS] - (A)$	$\phi_T = a_n - 1$		

V wird nicht berechnet.

Beispiele:

Nach der folgenden Tabelle wurde Dünger zu Pflanzen in drei separaten Schüben gegeben und das Wachstum der Pflanzen nach einer Woche gemessen. Von diesen Werten wollen wir die Ecksumme, den Freiraum, die objektive Veränderung und das Verhältnis der objektiven Veränderung ermitteln.

		Wachstum nach einer Woche (cm)					
Dünger	10	7.80	7.64	7.49	7.72	7.78b	
(g)	20	8.02	8.28	8.16	8.31	8.36	
	30	8.52	8.41	8.54	8.49	8.37	

Ausdruck:

Factor value

Replication value

3.

5.

\*Square sums

a=1.596

e=0.15956

t=1.75556

\*Degree of freedom

a=2.

e=12.

t=14.

\*Unbiased variance

a=0.798

e=1.329666667E-02

\*Unbiased variance ratio

a=60.01504135

X(1,1)=7,8

x(1,2)=7.64

X(1,3)=7.49

X(1,4)=7.72

X(1,5)=7.78

X(2,1)=8.02

X(2,2)=8.28

X(2,3)=8.16

X(2,4)=8.31

X(2,5)=8.36

X(3,1)=8.52

X(3,2)=8.41

X(3,3)=8.54

X(3,4)=8.49

X(3,5)=8.37

Tastenfolge

- |     |                              |                                |   |                                |   |
|-----|------------------------------|--------------------------------|---|--------------------------------|---|
| 1)  | <input type="checkbox"/> DEF | <input type="checkbox"/> A     | Programmstart   |                                |   |
| 2)  | 1                            | <input type="checkbox"/> ENTER | Ausgabe über den Drucker                              |                                |   |
| 3)  | 3                            | <input type="checkbox"/> ENTER | 5   | <input type="checkbox"/> ENTER | Eingabe der Zahl der Ebenen und<br>der Zahl der Abbildungen |
| 4)  | 7.8                          | <input type="checkbox"/> ENTER | Eingabe der Werte für X(1,1),<br>X(1,2) und<br>X(1,3) |                                |   |
|     | 7.64                         | <input type="checkbox"/> ENTER |   |                                |   |
|     | 7.58                         | <input type="checkbox"/> ENTER |   |                                |   |
|     |                              |                                |   |                                |   |
| 5)  | 8.54                         | <input type="checkbox"/> ENTER | Eingabe der Werte für X(3,3),<br>X(3,4) und<br>X(3,5) |                                |   |
|     | 8.49                         | <input type="checkbox"/> ENTER |   |                                |   |
|     | 8.37                         | <input type="checkbox"/> ENTER |   |                                |   |
| 6)  | <input type="checkbox"/> N   | <input type="checkbox"/> ENTER | Kein Ausdruck der Daten                               |                                |   |
| 7)  | <input type="checkbox"/> Y   | <input type="checkbox"/> ENTER | Korrektur der Daten                                   |                                |   |
| 8)  | 1                            | <input type="checkbox"/> ENTER | Eingabe der Spalte                                    |                                |   |
| 9)  | 3                            | <input type="checkbox"/> ENTER | Eingabe der Zeile                                     |                                |   |
| 10) | 7.49                         | <input type="checkbox"/> ENTER | Korrektur von X(1,3)                                  |                                |   |
| 11) | <input type="checkbox"/> N   | <input type="checkbox"/> ENTER | Keine weitere Korrektur                               |                                |   |
| 12) | <input type="checkbox"/> Y   | <input type="checkbox"/> ENTER | Ausdruck der Daten                                    |                                |   |
| 13) | <input type="checkbox"/> N   | <input type="checkbox"/> ENTER | Keine Korrektur, Programmende                         |                                |   |

## PROGRAMMLISTING

```

5 REM *****
10 REM SINGLE ARRANGE.
20 REM     METHOD
30 REM *****
40 "A": CLEAR : WAIT 0: CLS
50 BEEP 1
60 CURSOR 0,0: PRINT "*SINGLE ARRANGE. METHOD*"
70 CURSOR 2,2: GOTO 75
75 INPUT "PRINTER ON=1 OFF=2 ";P
80 IF (P<>1)*(P<>2) THEN 70
90 CURSOR 0,2: PRINT "
100 CURSOR 0,1: GOTO 105
105 INPUT "NO. OF FACTOR S=";A
110 CURSOR 0,2: GOTO 115
115 INPUT "NO. OF REPLICATIONS=";N
120 CURSOR
130 DIM D(A,N)
140 FOR I=1 TO A
150 FOR J=1 TO N
160 B$="X("+ STR$ (I)+", "+ STR$ (J)+")= "
170 CURSOR : PRINT B$
180 CURSOR LEN (B$),3: GOTO 190
190 INPUT D(I,J)
200 NEXT J
210 NEXT I
220 IF (P=2)*(F=0) THEN 370
230 IF (P=2)*(F=1) THEN 520
240 CURSOR :F=0: INPUT "PRINT OUT DATA(Y/N)";O$
250 IF O$="N" THEN 370
260 IF O$<>"Y" THEN 240
270 LPRINT : LPRINT "Factor value=": LPRINT A
280 LPRINT "Replication value=": LPRINT N: LPRINT
290 FOR I=1 TO A
300 FOR J=1 TO N
310 B$="X("+ STR$ (I)+", "+ STR$ (J)+")= "
320 LPRINT B$:D(I,J)
330 NEXT J
340 LPRINT
350 NEXT I
360 LPRINT
370 CURSOR : INPUT "AMEND DATA(Y/N)";O$
380 IF (O$="N")*(F=0) THEN 520
390 IF (O$="N")*(F=1) THEN 220
400 IF O$<>"Y" THEN 370
410 CLS :F=1
420 PRINT "X=           Y="
430 CURSOR 3,0:GOTO 435
435 INPUT X
440 IF (X<1)+(X>A) THEN 430
450 CURSOR 13,0: GOTO 455
455 INPUT Y: CURSOR
460 IF (Y<1)+(Y>N) THEN 450
470 B$="X("+ STR$ (X)+", "+ STR$ (Y)+")= "
480 PRINT B$;D(X,Y)
490 PRINT B$

```



```

500 CURSOR LEN (B$),2: GOTO 505
505 INPUT D(X,Y): CURSOR
510 GOTO 370
520 FOR I=1 TO A
530 E=0
540 FOR J=1 TO N
550 E=E+D(I,J)
560 Z=Z+D(I,J)*D(I,J)
570 NEXT J
580 S=S+E:T=T+E*E
590 NEXT I
600 S=S*S/(A*N):T=T/N:T=T-S:Z=Z-S:L=Z-T:F=A-1
610 U=T/F:O=A*(N-1):M=L/O:Q=A*N-1:G=U/M
620 ON P GOSUB 640,720
630 CLS : END
640 LPRINT "**Square sums"
650 LPRINT "a=";T:
    LPRINT "e=";L:
    LPRINT "t=";Z:LPRINT
660 LPRINT "**Degree of freedom"
670 LPRINT "a=";F:
    LPRINT "e=";O:
    LPRINT "t=";Q:LPRINT
680 LPRINT "**Unbiased variance"
690 LPRINT "a=";U:
    LPRINT "e=";M:
    LPRINT
700 LPRINT "**Unbiased variance ratio"
710 LPRINT "a=";G:
    LPRINT : LPRINT : RETURN
720 CLS : PRINT "**Square sums"
730 PRINT "a=";T: PRINT "e=";L: WAIT : PRINT "t=";Z: WAIT
740 WAIT 0: CLS : PRINT "**Degree of freedom"
750 PRINT "a=";F: PRINT "e=";O: WAIT : PRINT "t=";Q
760 WAIT 0: CLS : PRINT "**Unbiased variance"
770 PRINT "a=";U: WAIT : PRINT "e=";M
780 WAIT 0: CLS : PRINT "**Unbiased variance ratio"
790 WAIT : PRINT "a=";G
800 CLS : RETURN
    
```

SPEICHERINHALTE

A	Zahl der Ebenen	O	
B\$	Bildschirm	O\$	
E	Summe der Spaltenwerte	P	
F	Flag, Freiraum	Q	
G		S	Datensumme
I	Schleifenvariable	T	
J	Schleifenvariable	U	Varianz
L		X	Korrektur
M	Varianz	Y	Korrektur
N	Zahl der Abbildungen	D(A,N)	Daten

F-TEST

Konfiguration. CE-126P Drucker/Cassetten-Interface wird benötigt.

Überblick:

Eine Hilfe, um den F-Test schnell durchführen zu können.

Der F-Test ist eine große Hilfe, wenn festgestellt werden soll, ob ein Verhältnis zwischen den Bedingungen und den Resultaten eines Experiments bestehen.

Das Programm nimmt Ihnen schwierige, aber notwendige Berechnungen dieses Tests ab. Sie geben Datenzahl und Standardabweichungen oder Datenzahl und Daten ein, und der Rechner druckt Ihnen mit dem F-Test die Werte und den Freiraum aus.

Anweisungen:

Starten Sie das Programm mit DEF A, wenn Sie die Standardvariablen kennen.

1. Gegen Sie abhängig von der Bildschirmmaske die Datenzahl der Gruppe 1 und die Standardvariablen, die Datenzahl der Gruppe 2 und die Standardvariablen ein.
2. Sind die Eingaben vollständig, werden die Eingaben, der F-Test-Wert und der Freiraum ausgedruckt. Damit ist das Programm beendet.

Starten Sie das Programm mit DEF E, wenn Sie die Standardvariablen nicht kennen.

1. Geben Sie abhängig von der Bildschirmmaske die Datenzahl und die Daten der Gruppe 1, die Datenzahl und die Daten der Gruppe 2 ein.
2. Sind die Eingaben vollständig, werden die Eingaben, der Mittelwert, die Standardvariable, der F-Test-Wert und der Freiraum ausgedruckt. Damit ist das Programm beendet.

Inhalt:

\* Berechnung der Summe der Quadrate

Summe der Quadrate zwischen Klassen:  $\sum n_i (\bar{x}_i - \bar{x})^2$

Summe der Quadrate in der Klasse:  $\sum \sum (x_{ij} - \bar{x}_i)^2$

Gesamt:  $\sum \sum (x_{ij} - \bar{x})^2$

$\bar{x}_i$  : Mittelwert einer Klasse

$\bar{x}$  : Mittelwert aller Daten

## \* Berechnung des Freiraums

Freiraum zwischen Klassen: Zahl der Gruppen - 1

Freiraum in der Klasse: Gesamtsumme - Freiraum zwischen Klassen - 1

Gesamtsumme: Gesamtsumme der Daten - 1

## \* Berechnung der Quadrate der Mittelwerte

Quadrat des Mittelwerts zwischen Klassen:

Summe der Quadrate zwischen Klassen /  
Freiraum zwischen Klassen

Quadrat des Mittelwerts in der Klasse:

Summe der Quadrate in der Klasse /  
Freiraum in der Klasse

## \* Berechnung von F (Varianz-Verhältnis)

$F = \frac{\text{Quadrat des Mittelwerts zwischen Klassen}}{\text{Quadrat des Mittelwerts in der Klasse}}$

## Beispiele:

- Geben Sie die Werte der folgenden Tabelle ein und ermitteln Sie den Wert des F-Tests sowie den Freiraum.

Gruppe 1	Datenzahl : 10
	Standardvariable: 0.326114176
Gruppe 2	Datenzahl : 8
	Standardvariable: 0.356452736

2. Geben Sie die folgenden Daten ein und ermitteln Sie den Mittelwert, die Standardvariable, den F-Test und den Freiraum der einzelnen Gruppen.

	1	2	3	4	5	6	7
Gruppe 1	70	67	82	89	93	76	81
Gruppe 2	110	108	101	98	99	96	103

Ausdruck für Beispiel 1:

\*\*\*\*\* F TEST \*\*\*\*\*

Data numbers 1= 10.

Std. deviation 1= 0.326114176

Data numbers 2= 8.

Std. deviation 2= 0.356452736

F Test value= 1.1947115641

Degree of freedom 1= 7.

Degree of freedom 2= 9.

Ausdruck von Beispiel 2:

\*\*\*\*\* F TEST \*\*\*\*\*

Data numbers 1= 7.

Data 1(1)=70.

Data 1(2)=67.

Data 1(3)=82.

Data 1(4)=89.

Data 1(5)=93.

Data 1(6)=76.

Data 1(7)=81.

Average 1= 79.71428571

Std. deviation 1= 9.481812206

Data numbers 2= 7.

Data 2(1)=110.

Data 2(2)=108.

Data 2(3)=101.

Data 2(4)=98.

Data 2(5)=99.

Data 2(6)=96.

Data 2(7)=103.

Average 2= 102.1428571

Std. deviation 2= 5.209881704

---

F Test value= 3.312279483

Degree of freedom 1= 6.

Degree of freedom 2= 6.

Tastenfolge zu Beispiel 1:

- |    |                             |       |                           |
|----|-----------------------------|-------|---------------------------|
| 1) | DEF                         | A     | Programmstart             |
| 2) | 10                          | ENTER | Datenzahl Gruppe 1        |
| 3) | .326114176                  | ENTER | Standardvariable Gruppe 1 |
| 4) | 8                           | ENTER | Datenzahl Gruppe 2        |
| 5) | .356452736                  | ENTER | Standardvariable Gruppe 2 |
| 6) | AUSGABE ÜBER DEN<br>DRUCKER |       | Ausdruck, Programmende    |

Tastenfolge zu Beispiel 2:

- |    |                             |       |                                    |
|----|-----------------------------|-------|------------------------------------|
| 1) | DEF                         | B     | Programmstart                      |
| 2) | 7                           | ENTER | Datenzahl Gruppe 1                 |
| 3) | 70                          | ENTER | Eingabe des 1. Wertes der Gruppe 1 |
|    | :                           |       |                                    |
|    | :                           |       |                                    |
|    | :                           |       |                                    |
|    | :                           |       |                                    |
|    | :                           |       |                                    |
|    | :                           |       |                                    |
| 4) | 81                          | ENTER | Eingabe des 7. Wertes der Gruppe 1 |
| 5) | 7                           | ENTER | Datenzahl der Gruppe 2             |
|    | :                           |       |                                    |
|    | :                           |       |                                    |
|    | :                           |       |                                    |
|    | :                           |       |                                    |
|    | :                           |       |                                    |
| 6) | 96                          | ENTER | Eingabe des 6. Wertes der Gruppe 2 |
| 7) | 103                         | ENTER | Eingabe des 7. Wertes der Gruppe 2 |
| 8) | AUSGABE ÜBER DEN<br>DRUCKER |       | Ausdruck, Programmende             |

PROGRAMMLISTING

```

10 REM *****
20 REM *      F TEST      *
30 REM *****
40 "": CLEAR : CLS : WAIT 0
50 INPUT "DATA NUMBERS 1=";M
60 INPUT "STD. DEVIATION 1=";R
70 INPUT "DATA NUMBERS 2=";N
80 INPUT "STD. DEVIATION 2=";S
90 GOTO 330
100 "E": CLEAR : CLS : WAIT 0:F=1
110 INPUT "DATA NUMERBS 1=";M
120 DIM X(M)
130 PRINT ""
140 L=2
150 FOR I=1 TO M
160 CURSOR : PRINT "DATA 1("+ STR$ (I)+)"=";
170 CURSOR 9+ LEN ( STR$ (I)),L: GOTO 175
175 INPUT X(I)
180 X=X+X(I):R=R+X(I)*X(I)
190 IF L=4 LET L=3
200 NEXT I
210 X=X/M:R=R-M*X*X:R= SQR (R/(M-1))
220 CLS
230 INPUT "DATA NUMBERS 2=";N
240 DIM Y(N)
250 PRINT "":L=2
260 FOR I=1 TO N
270 CURSOR : PRINT "DATA 2("+ STR$ (I)+)"=";
280 CURSOR 9+ LEN ( STR$ (I)),L: GOTO 285
285 INPUT Y(I)
290 Y=Y+Y(I):S=S+Y(I)*Y(I)
300 L=L+1: IF L=4 LET L=3
310 NEXT I
320 Y=Y/N:S=S-N*Y*Y:S= SQR (S/(N-1))
330 T=R*R:U=S*S:O=M:Q=N
340 IF U>T LET P=M:M=N:N=P:P=U:U=T:T=P
350 LPRINT "***** F TEST *****"
360 LPRINT
370 LPRINT "Data numbers 1=": LPRINT O
380 IF F<>1 THEN GOTO 440
390 FOR I=1 TO O
400 LPRINT "Data 1("+ STR$ (I)+)"=";X(I)
410 NEXT I
420 LPRINT
430 LPRINT "Average 1=": LPRINT X
440 LPRINT "Std. deviation 1=": LPRINT R
450 LPRINT
460 LPRINT "Data numbers 2=": LPRINT Q
470 IF F<>1 THEN GOTO 530
480 FOR I=1 TO Q
490 LPRINT "Data 2("+ STR$ (I)+)"=";Y(I)
500 NEXT I
510 LPRINT
520 LPRINT "Average 2=": LPRINT Y
530 LPRINT "Std. deviation 2=": LPRINT S

```

```

540 LPRINT : LPRINT "-----": LPRINT
550 LPRINT " F Test value =": LPRINT T/U
560 LPRINT
570 LPRINT "Degree of freedom 1=": LPRINT M-1
580 LPRINT "Degree of freedom 2=": LPRINT N-1
590 LPRINT : LPRINT
600 END
    
```

Speicherinhalte

F		R	Standardvariable 1
I	Schleifenvariable	S	Standardvariable 2
L		T	Varianz 1
M	Datenzahl 1	U	Varianz 2
N	Datenzahl 2	X	Summe Daten 1, Quadrat
O		Y	Summe Daten 2, Quadrat
P		X(M)	Daten 1
Q		Y(N)	Daten 2



## ANHANG A

FEHLERMELDUNGEN

Fehlernummer	Bedeutung
1	<p>Syntax-Fehler</p> <p>Der PC 1350 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. (z. B. 5 / 0)</li> <li>3. Versucht, eine unlogische Rechnung ausführen. (z. B.: LN-30 oder ASN 1.5)</li> </ol>
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. (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. (z. B.: WAIT 66000)</p>

- 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 NEXT ohne FOR
- Das Einrichten einer Subroutine übersteigt 10 levels.
- Das Einrichten einer FOR-Schleife übersteigt 5 levels.
- Die Befehle RETURN ohne GOSUB, NEXT ohne FOR, READ ohne DATA wurden bereits 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. Sie kann aber auch auftreten,
- wenn ein Programm zu lang ist;
  - wenn der Reservespeicher mehr als 144 Bytes enthält.
- 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 PC 1350 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 PC 1350.

Andere Fehler

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.

EINGABEFEHLER.

Wenn Sie ein Programm ausführen, kann eine Fehlermeldung aufgrund eines Eingabefehlers auftreten. Beachten Sie in diesem Fall die folgenden Anmerkungen:

Beispiel:

Sie haben KPRINT anstelle von LPRINT eingegeben.

EINGABE

ANZEIGE







 L 






10: K PRINT A\$  
 10  PRINT A\$  
 10: L PRINT A\$  
 ↙  
 Space

Wenn Sie auf diese Weise die Korrektur durchführen, versteht der Computer sie nicht als Befehl. In diesem Beispiel müssen Sie KPRINT löschen und dann LPRINT neu eingeben.

EINGABE




ANZEIGE





  
   
 ...   
 LPRINT 

10: K PRINT A\$  
 10  PRINT A\$  
 10 \$  
 10     A\$  
 10: LPRINT A\$  
 ↙  
 Kein Space, da als Befehl erkannt.

Mit der Cursor-Taste können Sie den Befehl auf korrekte Eingabe überprüfen.

RICHTIGE EINGABEFALSCHE EINGABE

 10: RADIAN  
 10 RADIAN  
 10 RADIAN\_

10: RADAN  
 10 RADAN  
 10 RADAN  
 10 RADAN  
 10 RADAN

ANHANG B  
ASCII-CODE TABELLE

Entsprechender Code

ASCII Zeichen	Dual	Dezimal	Hexadezimal
Leerraum	00100000	32	20
!	00100001	33	21
"	00100010	34	22
#	00100011	35	23
\$	00100100	36	24
%	00100101	37	25
&	00100110	38	26
'	00100111	39	27
(	00101000	40	28
)	00101001	41	29
*	00101010	42	2A
+	00101011	43	2B
,	00101100	44	2C
-	00101101	45	2D
.	00101110	46	2E
/	00101111	47	2F
0	00110000	48	30
1	00110001	49	31
2	00110010	50	32
3	00110011	51	33
4	00110100	52	34
5	00110101	53	35
6	00110110	54	36
7	00110111	55	37
8	00111000	56	38
9	00111001	57	39
:	00111010	58	3A
;	00111011	59	3B
<	00111100	60	3C
=	00111101	61	3D
>	00111110	62	3E
?	00111111	63	3F
@	01000000	64	40

ASCII Zeichen	Entsprechender Code		
	Dual	Dezimal	Hexadezimal
A	01000001	65	41
B	01000010	66	42
C	01000011	67	43
D	01000100	68	44
E	01000101	69	45
F	01000110	70	46
G	01000111	71	47
H	01001000	72	48
I	01001001	73	49
J	01001010	74	4A
K	01001011	75	4B
L	01001100	76	4C
M	01001101	77	4D
N	01001110	78	4E
O	01001111	79	4F
P	01010000	80	50
Q	01010001	81	51
R	01010010	82	52
S	01010011	83	53
T	01010100	84	54
U	01010101	85	55
V	01010110	86	56
W	01010111	87	57
X	01011000	88	58
Y	01011001	89	59
Z	01011010	90	5A
[	01011011	91	5B
\	01011100	92	5C
]	01011101	93	5D
^	01011110	94	5E
_	01011111	95	5F
?	01100000	96	60
a	01100001	97	61
b	01100010	98	62
c	01100011	99	63
d	01100100	100	64
e	01100101	101	65
f	01100110	102	66
g	01100111	103	67
h	01101000	104	68
i	01101001	105	69

ASCII Zeichen	Entsprechender Code		
	Dual	Dezimal	Hexadezimal
j	01101010	106	6A
k	01101011	107	6B
l	01101100	108	6C
m	01101101	109	6D
n	01101110	110	6E
o	01101111	111	6F
p	01110000	112	70
q	01110001	113	71
r	01110010	114	72
s	01110011	115	73
t	01110100	116	74
u	01110101	117	75
v	01110110	118	76
w	01110111	119	77
x	01111000	120	78
y	01111001	121	79
z	01111010	122	7A
{	01111011	123	7B
:	01111100	124	7C
}	01111101	125	7D
~	01111110	126	7E
▲	11110101	245	F5
▼	11110110	246	F6
◆	11110111	247	F7
♣	11111000	248	F8
♠	11111001	249	F9
♣	11111010	250	FA
♠	11111011	251	FB
√	11111100	252	FC

Achtung: Die Zeichencodes 92 (&5C), 249 (&F9) und 250 (&FA) des PC 1350 unterscheiden sich von denen des Druckers CE-126P (Ausdruck).

Verwenden Sie nicht den Zeichencode Ø (&Ø), wenn Sie den Drucker angeschlossen haben.

## ANHANG C

### FORMATIEREN DER DATENAUSGABE

Es ist manchmal wichtig oder nützlich, neben dem Inhalt ausgegebener Daten auch das Format zu kontrollieren. Der PC 1350 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.

```
10: USING "####"
20: M$ = "&&&&"
30: USING M$
```

Wird der Befehl USING ohne Maske benutzt, werden damit alle Spezialformate aufgehoben.

```
40: USING
```

Der Befehl USING kann auch in einem PRINT-Statement benutzt werden.

```
50: 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 PC 1350 nicht mehr als 10 Stellen erfassen kann, sollte die größte verwendete numerische Maske maximal 11 "#"-Zeichen enthalten.

Wenn die Gesamtstellenzahl 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.

EINGABEANZEIGE

10: USING "####"

(Bringen Sie den PC 1350 in den RUN-Mode, geben Sie RUN ein und drücken Sie die ENTER -Taste.)

20: PRINT 25

| 25|

30: PRINT -350

| -350|

40: PRINT 1000

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 anzuzeigenden Wert benötigt werden, werden die auf der rechten Seite übrigbleibenden Stellen mit Nullen aufgefüllt. Enthält der anzuzeigende Wert mehr Stellen, als die Maske vorsieht, werden diese abgeschnitten (nicht gerundet).

EINGABEANZEIGE

10: USING "###.#.###"

20: PRINT 25

| 25.00|

30: PRINT -350.5

| -350.50|

40: PRINT 2.547

| 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 "E" 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 "E"-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 "E"-Zeichen). Nach dem "charakteristischen" Teil wird das Exponentiationszeichen (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 "E^", damit werden Zahlen in der Form "2 E 99" dargestellt. Das größtmögliche Feld für wissenschaftliche Notation wäre "E.E", das Zahlen wie "-1.234567890 E -12" darstellen kann.

EINGABE

ANZEIGE

```
10: USING "E.E^"
20: PRINT 2           | 2.00 E 00 |
30: PRINT -365.278   |-3.65 E 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

ANZEIGE

```
10: USING "&&&&&"
20: PRINT "ABC"       |ABC   |
30: PRINT "ABCDEFGHI" |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 "E.E&&&" zwei verschiedene Werte formatiert werden - ein numerischer Ausdruck, für den 5 Stellen vorgesehen sind, und ein alphanumerischer Ausdruck, für den 4 Stellen bereitgestellt wurden.

EINGABE

10: PRINT USING "###.##&";25;"CR"  
20: PRINT -5.789;"DB"

ANZEIGE

|25.00CR|  
|-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.

## ANHANG D

## BEWERTUNG VON AUSDRÜCKEN UND OPERATOREN-VORRANG

Wenn in den SHARP PC 1350 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 PC 1350 nicht, ob Sie meinen:

$$\frac{100}{5 + 45} = 2 \quad \text{oder} \quad \frac{100}{5} + 45 = 65$$

Da der PC 1350 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 vom SHARP PC 1350 gemäß den folgenden Prioritätsregeln verarbeitet (angefangen mit der höchsten Priorität):

1. Klammern
2. Variable und Pseudovariablen
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.:

$$-2^4 \longrightarrow -(2^4)$$

$$3^{-2} \longrightarrow \frac{3}{3^2}$$

#### BEISPIEL FÜR EINE BEWERTUNGSFOLGE

Wir gehen aus von dem Ausdruck:

$$((3+5-2)*6+2)/10^{\text{LOG } 100}$$

Der PC 1350 verarbeitet nun zuerst die innersten Klammern. Da "+" und "-" auf derselben Stufe stehen, wird von links nach rechts gerechnet, also die Addition zuerst ausgeführt.

$$((8-2)*6+2)/10^{\text{LOG } 100}$$

Dann wird die Subtraktion durchgeführt:

$$((6)*6+2)/10^{\text{LOG } 100}$$

Oder:

$$(6*6+2)/10^{\text{LOG } 100}$$

In der nächsten Klammer wird zuerst die Multiplikation durchgeführt:

$$(36+2)/10^{\text{LOG } 100}$$

Dann folgt die Addition:

$$(38)/10^{\text{LOG } 100}$$

Oder:

$$38/10^{\text{LOG } 100}$$

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 PC 1350 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 Befehlsausführung unterbrochen.
- SHIFT** Die gelbe Taste mit der Aufschrift "SHIFT" muß benutzt werden, um Doppelaktionen 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 Anzeigeeinhalt, 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.
- MODE** Mit dieser Taste können Sie den Modus (RUN oder PROGRAMN) auswählen.
- SHIFT MODE** Mit dieser Tastenkombination schalten Sie den RESERVE-Modus ein.
- 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 Buchstabentaste "E").
- /** Divisionszeichen
- \*** Multiplikationszeichen
- wird benutzt, um eine Feld-Variable in INPUT# , PRINT# etc. zu bestimmen.
- +** Additionszeichen
- Subtraktionszeichen
- SHIFT** **^** Wird benutzt, um Zahlen zu potenzieren und um das Exponenten-Anzeige-System für numerische Daten in USING-Anweisungen zu spezifizieren.
- SHIFT** **<**  
**SHIFT** **>** Werden benutzt, wenn logische Operatoren in einer IF-Sequenz eingegeben werden sollen.
- DEF** 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 **DEF**-Taste gedrückt wird, wird das Programm mit dem entsprechenden Kennbuchstaben gestartet.
- A** bis **Z** 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 **SML** drücken.
- SPC** Wird beim Eingeben von Zeichen oder Programmen benutzt, um einen Leerschritt zu produzieren.
- =**
- 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.



- SHIFT** !"#\$\$  
%&@
- 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 Buchstabeinhalte gegeben;  
 - weist hexadezimale Zahlen aus;  
 @ - wird für den Reservespeicher benutzt, wenn die Reservetaste als Programmtaste 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.
- [ ( ]
- Zur Eingabe von Klammern
- [ ) ]
- [▶]
- Bewegt den Cursor nach rechts (auf einmaligen Tastendruck Bewegung um eine Stelle; wird die Taste festgehalten, Dauerfunktion).  
 - 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, Dauerfunktion).  
 - Sonst wie unter der [▶]-Taste beschrieben.
- [INS]
- Fügt einen Leerschritt ein, dabei erscheint das Zeichen "□". Dieser Leerschritt wird vor das Zeichen gesetzt, auf dem der Cursor steht.
- [DEL]
- Löscht den Inhalt der Stelle, auf der der Cursor steht.
- SHIFT** [π]
- Zur Eingabe von Pi (π)
- SHIFT** [√]
- Zur Eingabe der Quadratwurzel

**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 ein INPUT- oder PRINT-Befehl unterbrochen wurde.



**SHIFT****P↔NP**

Stellt die Verbindung zum Drucker her bzw. unterbindet sie (sofern ein zusätzlicher Drucker an den PC 1350 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		
RUN	Programmdurchführung		
	Programm vorübergehend unterbrochen,	um die nächste Zeile durchführen zu lassen.	Diese Taste festhalten, um bereits abgearbeitete Zeilen zur Anzeige zu bringen.
	INPUT-, PRINT-, LINE-, PRESET-, PSET-, GPRINT-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.
RESERVE			

- In der Anzeige ist die **ENTER** -Taste das gleiche wie ein Leerschritt.
- Wenn 11 minuten lang keine Taste betätigt wird, schaltet sich der Computer automatisch ab.

## ANHANG F

## SYMBOLS DES E/A-ANSCHLUSSES

Der PC 1350 ist mit einem 15poligen Anschluß für ein E/A-Terminal ausgestattet. Die Anschlüsse und ihre Symbole werden anschließend beschrieben.



Anschluß	Name	Symbol	E/A	Funktion
2	Transmit Data	SD	o	DC Ausgangssignal
3	Receive Data	RD	i	DC Eingangssignal
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
11	Receive Ready	RR	o	ON: Empfang möglich
14	Data Terminal Ready	DR	o	ON: Terminal bereit
1	Frame Ground	FG		Massenanschluß für Wartung
10		VC		Energieversorgung
13				

**Achtung:** Das Spannungsniveau von VC entspricht der logischen 1, das Spannungsniveau von SG der logischen 0.

**Achtung:** wenn der PC 1350 an eine Spannung angeschlossen wird, die das erlaubte Niveau überschreitet (Spannungsniveau zwischen SG und VC), kann es passieren, daß die internen Bausteine des PC 1350 beschädigt werden, da der Computer CMOS-Bauelemente besitzt.

## ANHANG G

## TECHNISCHE DATEN

Modell:	PC 1350 Taschencomputer
Prozessor:	8 Bit CMOS CPU
Programmiersprache:	BASIC
Speicherkapazität:	System-ROM 40 K Bytes
	RAM:
	System: 944 Bytes
	Anwender: Fester Speicherbereich
	208 Bytes (A-Z, A\$-Z\$)
	Programm-/Datenbereich
	3070 Bytes
	Reservespeicher 144 Bytes
Stack:	Unterroutine: 10 Stacks
	Funktion: 16 Stacks
	FOR-NEXT: 5 Stacks
	Daten: 8 Stacks
Operatoren:	Addition, Subtraktion, Multiplikation, Division, trigonometrische und invers-trigonometrische 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, Zeicheneinfügung, Zeichenlöschung
Speicherschutz:	CMOS, batteriegestützt
Serieller E/A-Anschluß und seine Möglichkeiten:	
Standards:	Start-Stop-Übertragungssystem (asynchron). Nur Halb-Duplex.
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 Ausgaben: SD, RS, RH, ER Andere: SG, FG, VC
Anzeige:	Vierzeilige 24-Stellen-Flüssigkristall-anzeige mit 5x7-Punkt-Zeichen oder Graphik mit 150 x 32 Punkten.
Tastatur:	62 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. 250 Stunden, wenn ständig 5555555555 angezeigt wird bei einer Betriebstemperatur von 20 Grad Celsius. Diese Betriebszeit kann leicht schwanken, in Abhängigkeit von Arbeitsmethoden etc. * Bei einständigem 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. 190 g (mit Batterien)
Zubehör:	Deckel, 2 eingebaute Lithium-Zellen, zwei Tastaturschablonen und ein Handbuch
Zusatzgeräte:	Einsteckkarte - 8 K Bytes RAM (CE-201M) - 16 K Bytes RAM (CE-202M) Cassettenrecorder (CE-152) Drucker/Cassetten-Interface (CE-126P) etc.

## ANHANG H

DIE BENUTZUNG VON PROGRAMMEN, DIE FÜR DEN PC 1245,  
PC 1250/1251 ODER PC 1401 ENTWICKELT WURDEN

Achtung: PC 1210 Serien: PC 1210, PC 1211  
 PC 1245 Serien: PC 1245, PC 1250, PC 1251, PC 1255  
 PC 1260 Serien: PC 1260, PC 1261

Die Anzeigetafeln bei den Serien PC 1245, PC 1210, PC 1260 und den PC 1401 Taschencomputern bestehen entweder aus 1 oder 2 Zeilen. Deshalb müssen Sie, wenn Sie den Bildschirminhalt "wechsein" wollen, zuvor den angezeigten Text löschen. Beim PC 1350 rollt der Bildschirminhalt dagegen mit jeder neuen Zeile einfach nach oben.

Aus diesem Grunde können bei der Programmausführung in der PC 1260 Serie oder bei Programmen mit dem CURSOR-Kommando unerwünschte Bewegungen oder Bildschirmanzeigen erfolgen. Es ist daher notwendig, das Programm abzuändern, indem Sie z. B. den Bildschirminhalt mit dem CLS-Kommando löschen.

Es kann außerdem notwendig sein, andere kleine Abänderungen entsprechend dem Typ Ihres Taschencomputers vorzunehmen. Diese Modifikationen sind im folgenden beschrieben.

MODIFIKATIONEN VON PROGRAMMEN DER PC 1245 SERIE (PC 1245, PC 1250,  
PC 1251)

Wenn Sie die Programme, die für die PC 1245 Serie entwickelt wurden, auf dem PC 1350 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 1350 nicht möglich, da dieser zwei aufeinanderfolgende Zeichen als einfache Variable behandelt. Benutzen Sie daher die Schreibweise rechts im Beispiel:

(z. B.) A = SIN BC  $\longrightarrow$  A = 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 1350 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 angegebenen Schreibweise:

(z. B.) DIM A(30)  $\longrightarrow$  A(30) = 0



### 3. Datenein-/ausgabebefehl 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 1350 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 1350 von dem bei der PC 1245 Serie. Wird der Wert einer Schleifenvariablen in einem Bedingungsdruck benutzt, müssen Sie ihn beim PC 1350 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 100
```

(Auf dem PC 1350 müssen Sie den Wert einer Schleifenvariablen um eins erhöhen. Die Anzahl der Schleifendurchläufe bleibt jedoch gleich.)

### 5. Exponentialzeichen "E"

Der PC 1350 benutzt den Großbuchstaben "E" für die Darstellung der Exponentiation. Die folgenden Änderungen werden dadurch erforderlich:

```
A = 1.234 E 5  → A = 1.234E5
B = E 6       → B = 1E6
```

Wird ein Programm der PC 1245 Serie von Band in den PC 1350 eingelesen, wird diese Veränderung automatisch vom PC 1350 durchgeführt.)

6. Der Zeichen-Code der PC 1245 Serie unterscheidet sich geringfügig von dem des PC 1350. Wenn die folgenden Codes durch die CHR\$-Funktion zugewiesen werden, ändern Sie die Codes:

ASCII	PC 1245	PC 1350
39 (&27)	⌈	,
91 (&5B)	⌋	⌈
92 (&50)	⌘	\
93 (&5D)	⌞	⌋
96 (&60)	⌚	,
251 (&FB)	- (ERROR)	⌞
252 (&FC)	- (ERROR)	⌋

Anmerkung: Wie oben gezeigt, verfügt der PC 1350 nicht über das Zeichen ⌚.

ZUSÄTZLICHE MODIFIKATIONEN

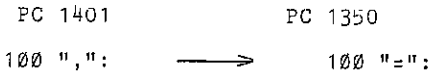
1. Die Modelle PC 1245, PC 1250 und PC 1251 verfügen über eine Programmzeilenspanne von 1 - 999. Der PC 1350 hat eine erweiterte Spanne von 1 - 65279. Aus diesem Grund beansprucht die Zeilennummer 3 Bytes im RAM (bei der PC 1245 Serie nur 2 Bytes.). Die Änderung wird automatisch ausgeführt, wenn Sie Programme von Cassette in den Computer einladen. Es besteht die Möglichkeit eines Speicherüberlaufs, wenn Sie zu lange Programme laden oder ausführen lassen (ERROR 6).
2. Wenn eine Zeile etwa 80 Bytes umfaßt, kann durch diese Veränderung das Zeilenende gelöscht werden.
3. Wenn Sie ein Programm der PC 1245 Serie in den PC 1350 einladen, wird aufgrund der Änderung in der Zeilennummerierung (von 2 auf 3 Bytes) die BUSY-Anzeige noch für einige Sekunden aufleuchten, nachdem das Band zu laufen aufgehört hat. Während dieser Zeit wird das "\*" -Zeichen in der rechten unteren Ecke des Zeilendisplays sichtbar sein.

ÄNDERUNGEN BEZÜGLICH DES PC 1401

1. Änderungen der definierbaren ", "-Taste

bei dem PC 1350 ist die ", "-Taste keine definierbare Taste. Aus diesem Grund müssen Sie, wenn in einem Programm eine Taste definiert werden muß, eine andere wählen.

Beispiel:



## 2. Andere Anmerkungen

Der PC 1401 verfügt über einige Funktionskommandos mehr als der PC 1350. Darum wird bei der Programmausführung immer dann eine Fehlermeldung auftauchen, wenn ein Programm geschrieben oder eingelesen wird, das Befehle enthält, die der PC 1350 nicht verarbeiten kann.

Wird ein Programm mit solchen Befehlen von Band eingelesen, werden diese Befehle durch "~" ersetzt und dann in der Anzeige ausgegeben.

ÄNDERUNGEN BEZÜGLICH DES PC 1260/1261

## 1. Leerzeichen:

Bei der PC 1260 Serie ist der Zeichencode 96 (&60) gleich ein Leerzeichen. Dieses Zeichen wird auf dem PC 1350 als "c" dargestellt. Ändern Sie deshalb den Code 96 bei der CHR\$-Funktion in 32 (&20) um.

## 2. INPUT-Befehl:

Wenn in der PC 1260 Serie ein INPUT-Befehl mit einer Anzeige-Startposition (CURSOR-Befehl) verwendet wurde, so ist für den PC 1350 ein GOTO-Befehl hinzuzufügen, wie in den folgenden Beispielen gezeigt ist.

## Beispiel 1

```

.
.
.
50 CURSOR 24
60 INPUT "A= ";A
.
.

```



```

.
.
50 CURSOR 24
55 GOTO 60
60 INPUT "A= ";A
.
.

```

Fügen Sie diesen GOTO-Befehl ein.

## Beispiel 2

```

.
.
80 CURSOR 31 : INPUT "B= ";B
.
.
↓
.
.
80 CURSOR 31 : GOTO 85
85 INPUT "B= ";B
.
.

```

ÄNDERUNGEN BEZÜGLICH DER PC 1210 SERIE

Um Programm der PC 1210 Serie auf dem PC 1350 zu benutzen, müssen diese in gleicher Weise geändert werden, wie Programme der PC 1245 Serie (außer die Punkte 2. und 6.). 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 1350 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 1350.

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 1350 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	-123.45 -123
-----------	--	------	-----------------

* PC 1350		-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 1350 gestaltet sich die Anzeige entsprechend dem vorangegegangenen 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 1350 kann das ")"-Zeichen nicht ausgelassen werden.

Deshalb müssen Sie es einfügen, wenn Sie Programme übernehmen wollen.

## 4. PRINT-Befehl

Der PC 1350 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 1350 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 1350 werden alle Variablen von A(27) an gelöscht.

Deshalb ist es nötig, falls die Variablen beibehalten werden sollen, das Programm auf dem PC 1350 mittels GOTO oder der DEF-Taste zu starten.









**SHARP CORPORATION**

**OSAKA, JAPAN**

1983 © SHARP CORPORATION  
Printed in Japan/Imprimé au Japon

4K3.2T(TINSG4221CCZZ)①