

HANDY PERSONAL COMPUTER  
**CASIO FP-200**  
**KOMMANDO-LISTE**

**CASIO®**



**CASIO®**

## BASIC PF Tasten-Liste

PF-Taste	Funktion
PF0	Zur Anzeige von "EDIT". Dient zur Anzeige der spezifizierten Zeile und zum Aktivieren der Edit-Betriebsart.
PF1	Anzeige von "PROG". Dient zur Modifikation von Programmbereichen usw.
PF2	"SYSTEM C/R". Anzeige des Zustands der Programmbereich-Verwendung.
PF3	"LIST C/R". Vollständige oder teilweise Anzeige eines Programms.
PF4	"RUN C/R". Ausführung eines Programms.
PF5 (SHIFT+PF0)	Umschaltung zwischen NUM-Betriebsart und Normal-Betriebsart.
PF6 (SHIFT+PF1)	"FILES C/R". Anzeige der Namen und Attribute aller Dateien auf einer Floppy Disk.
PF7 (SHIFT+PF2)	Anzeige von "LOAD". Dient zum Laden eines Programms in den Speicher.
PF8 (SHIFT+PF3)	Anzeige von "SAVE" Dient zum Speichern eines Programms in der spezifizierten Datei.
PF9 (SHIFT+PF4)	"P.DATE\$, TIME\$ C/R". Anzeige von Datum und Uhrzeit.

## CETL PF Tastenliste

PF-Taste	Funktion
PF0	Umschaltung zwischen Edit-Betriebsart und Kommando-Betriebsart.
PF1	Zeigt "FILE" an. Wird zur Umschaltung von Dateibereichen usw. verwendet.
PF2	"SYSTEM C/R" zeigt den Status des verwendeten Dateibereichs an.
PF3	Zeigt das Kommando-Menü an.
PF4	"P. FRE C/R" zeigt die Größe des unbenutzten CETL-Bereichs an.
PF5(SHIFT + PF0)	Umschaltung zwischen NUM-Betriebsart und Normal-Betriebsart. (Die NUM-Betriebsart verwendet einige Buchstaben-Tasten als zusätzliche numerische Tasten).
PF6(SHIFT + PF1)	Zeigt "FL(" an.
PF7(SHIFT + PF2)	Zeigt "RC(" an.
PF8(SHIFT + PF3)	Zeigt "IT(" an.
PF9(SHIFT + PF4)	"P.DATES, TIMES/C/R". Zur Anzeige von Datum und Uhrzeit.

## CETL-Kommandoliste

Kommando	Funktion
N	Schafft eine neue Tabelle (gibt den Dateinamen, die Anzahl der Sätze und Anzahl der Posten ein).
A	Legt die Eingaberichtung fest.
I	Zum Einfügen von Sätzen oder Posten.
D	Zum Löschen von Sätzen oder Posten.
M	Zur Bewegung (Verschiebung von Sätzen oder Posten).
B	Zum Einfügen von Nullen oder Leerstellen in eine oder mehrere Datenzellen.
S	Zum Sortieren der angegebenen Posten in aufsteigender oder absteigender Reihenfolge und Ausgabe der Ergebnisse.
F	Suchfunktion mit vorgegebener Bedingung.
J	Sprung von der gegenwärtigen Datenzelle zu einer anderen Zelle.
C	Führt eine Rechnung aus und bewahrt das Ergebnis in der entsprechenden Datenzelle.
T	Zur Ausgabe von Daten im Tabellenformat auch über die Anzeige oder den Drucker.
R	Änderung des Namens der zur Zeit aktivierten Datei.
K	Löscht den zur Zeit aktivierten Dateibereich oder den gesamten Dateibereich.
P	Ausgabe von Daten über ein Peripheriegerät.
G	Eingabe von Daten von einem Peripheriegerät.
L	Auflisten des Inhalts einer Datei über den Drucker.



## CETL Management-Funktionen

Funktion	Ergebnis
RC	Anzeige der gegenwärtigen Satznummer.
IT	Anzeige der gegenwärtigen Postennummer.
RC (Satznummer)	Anzeige des Wertes der Datenzelle im zur Zeit aktivierten Posten mit der spezifizierten Satznummer.
IT (Satznummer)	Anzeige des Wertes der Datenzelle im derzeit aktivierten Satz mit der spezifizierten Postennummer.
FL ([Dateinummer] , [Satznummer] , [Postennummer] )	Anzeige des Wertes der Datenzelle bei Spezifikation von Dateibereich, Satznummer und Postennummer.
SUMRC(Satznummer m , Satznummer n)	Anzeige der Summe der Zellenwerte im derzeit aktivierten Posten mit den Satznummern (einschließlich) m bis n.
SUMIT (Postennummer m, Postennummer n)	Anzeige der Summe von Zellenwerten im zur Zeit aktivierten Satz mit den Postennummern (einschließlich) m bis n.

\* Ein Dateiname, ein Satzname, der Inhalt der Datenzellen von Posten 1 oder ein Postenname kann jeweils anstelle der Dateinummer, Satznummer oder Postennummer als Parameter verwendet werden.

## System-Kommandos

C-85 BASIC macht keine Unterscheidungen zwischen Kommandos und Anweisungen. Die hauptsächlich in der Kommando-Betriebsart verwendeten Anweisungen nennen wir Kommandos, und die in Programmen verwendeten nennen wir Anweisungen. Die durch [ ] eingeschlossenen Elemente sind optional. Elemente mit einer \*-Markierung in der oberen rechten Ecke können wiederholt verwendet werden. Die F-Markierung rechts von einem Kommando zeigt an, daß das Kommando eine Funktion ist.

### AREA

**Funktion** Legt die Größe des CETL-Bereiches fest.  
**Format** AREA numerischer Ausdruck.  
**Beispiel** AREA 1500

### CLEAR

**Funktion** Löscht alle Variablen und vom Benutzer gewählten Funktionen und spezifiziert die Größe des Zeichenbereichs und die obere Grenze des BASIC-Bereichs.  
**Format** CLEAR [ "Größe des Zeichenbereichs [ , obere Grenze des BASIC-Bereichs ] ].  
 \* Die Größe des Zeichenbereichs wird nach Rückstellung auf 1023 eingestellt.  
**Beispiel** CLEAR CL

### EDIT

**Funktion** Zeigt die spezifizierte Zeile an und initiiert die Edit-Betriebsart.  
**Format** EDIT [ Zeilennummer ]  
**Beispiel** EDIT 20

### KEY

**Funktion** Definiert eine programmierbare Funktionstaste  
**Format** KEY Funktionstastennummer, Zeichenausdruck  
**Beispiel** KEY 3, "LIST" + CHR\$ (13)

### KEY LIST

**Funktion** Anzeige der Definitionen gegenwärtig aktiver, programmierbarer Funktionstasten (PF).  
**Format** KEY LIST  
**Beispiel** KEY LIST

### NEW

**Funktion** Löscht ein Programm  
**Format** NEW [ALL]  
 \* NEW ALL löscht alle Programme und Variablen  
**Beispiel** NEW

### LIST

**Funktion** Anzeige des gesamten oder eines Teils des Programms.  
**Format** LIST [erste Zeilennummer] [ | \_ | letzte Zeilennummer ] ]

**Beispiele** LIST  
LIST, 40  
LIST, 20, 50  
LIST 40,

#### PASS

**Funktion** Anweisung bzw. Aufhebung eines Paßwortes  
**Format** PASS Zeichenkonstante  
**Beispiel** PASS "ORG"

#### PROG

**Funktion** Wahl des Programmbereiches  
**Format** PROG Programmbereich-Nummer  
**Beispiel** PROG 3

#### RENUM

**Funktion** Neunumerierung von Programmzeilen  
**Format** RENUM [neue Zeilennummer] [, [alte Zeilennummer] [, [Inkrement ]]]  
**Beispiel** RENUM 100, 10, 15

#### RESET

**Funktion** Initialisiert den RAM-Arbeitsspeicher  
**Format** RESET  
**Beispiel** RESET

#### RUN

**Funktion** Ausführung eines Programms  
**Format** RUN [Zeilennummer]  
**Beispiel** RUN 100

#### SYSTEM

**Funktion** Anzeige des Zustands der Programmbereiche und Dateibereiche  
**Format** SYSTEM  
**Beispiel** SYSTEM

#### TRON

**Funktion** Verfolgung eines Programmablaufs  
**Format** TRON  
**Beispiel** TRON

#### TROFF

**Funktion** Beendet die Verfolgung einer Programmausführung  
**Format** TROFF  
**Beispiel** TROFF

#### • Ausführung-Steueranweisungen

#### FOR-NEXT

**Funktion** Wiederholt die Ausführung der Anweisungen zwischen der FOR-Anweisung und der NEXT-Anweisung eine festgelegte Anzahl Male

**Format** FOR Variablenname = Ausgangswert TO letzter Wert [STEP Zunahme]  
NEXT [Variablenname [, Variablenname]]  
**Beispiel** FOR I=1 TO 12 STEP 3  
NEXT I

#### END

**Funktion** Beendet die Programmausführung  
**Format** END  
**Beispiel** END

#### GOSUB-RETURN

**Funktion** Bewirkt einen Sprung in ein Unterprogramm und Rückkehr aus dem Unterprogramm  
**Format** GOSUB Zeilennummer  
RETURN PROG Programmbereich-Nummer  
**Beispiel** GOSUB 50

#### GOTO

**Funktion** Bewirkt die Verzweigung zum spezifizierten Ziel  
**Format** GOTO { Ziel-Zeilennummer  
PROG Programmbereich-Nummer }  
**Beispiel** GOTO 100

#### IF-THEN-ELSE

**Funktion** Führt eine bedingte Verzweigung je nach der angegebenen Verzweigungsbedingung durch.  
**Format** IF Verzweigungsbedingung Ausdruck THEN  
{ Anweisung [ : Anweisung ] \* }  
{ Verzweigungsziel }  
{ ELSE Anweisung { [ : Anweisung ] \* } }  
{ [ Zeilennummer ] }  
\* Das Verzweigungsziel ist:  
{ Ziel-Zeilennummer }  
{ PROG Programmbereich-Nummer }  
**Beispiel** IF A > 0 THEN A=1 ELSE A=0

#### ON-GOSUB/ON-GOTO

**Funktion** Bewirkt abhängig vom Wert eines numerischen Ausdrucks den Sprung zu einer von mehreren spezifizierten Ziellinien  
**Format** ON Numerischer Ausdruck  
{ GOSUB [Zeilennummer] [, Zeilennummer]. . . }  
{ GOTO [Zeilennummer] [, Zeilennummer]. . . }  
**Beispiele** ON I GOSUB, 100,, 200,, 300,, 400  
ON I GOTO, 50, 60, 70,

#### STOP

**Funktion** Die Programmausführung wird kurzzeitig gestoppt und das System wartet auf die nächste Kommandoeingabe  
**Format** STOP  
**Beispiel** STOP

**RETURN**

**Funktion** Bewirkt die Rückkehr aus einem Unterprogramm  
**Format** RETURN  
**Beispiel** RETURN

• **Kommentar-Anweisung****REM**

**Funktion** Zur Aufnahme von Anmerkungen im Programm  
**Format** REM Anmerkung  
**Beispiel** REM SUMMATION OF 1 TO 1000

• **Datenveränderung und Verkettung****LET**

**Funktion** Ordnet einer Variablen einen Wert zu.  
**Format** [LET] numerischer Variablenname = numerischer Ausdruck  
 Zeichen Variablenname = Zeichen - Ausdruck  
**Beispiele** LET X=12  
 LET Y=X\*X+2\*X-1  
 \* Das Wort LET kann auf Wunsch auch ausgelassen werden.

• **Lesen von Daten innerhalb eines Programms****DATA**

**Funktion** Spezifiziert die mit einer READ-Anweisung gelesenen numerischen Werte oder Zeichenkonstanten  
**Format** DATA Konstante [, Konstante ... ] \*  
**Beispiel** DATA 2, 3, START, END

**READ**

**Funktion** Liest eine Konstante von einer DATA-Anweisung in die spezifizierte Variable  
**Format** READ Variablenname [, Variablenname ... ] \*  
**Beispiel** READ A, B, C,

**RESTORE**

**Funktion** Spezifizieren der Zeilennummer der von der nächsten READ-Anweisung zu lesenden DATA-Anweisung.  
**Format** RESTORE [Zeilennummer]  
**Beispiel** RESTORE 10

• **Anzeige****CLS**

**Funktion** Löscht die Anzeige  
**Format** CLS  
**Beispiel** CLS

**LOCATE**

**Funktion** Bewegt den Cursor zu einer spezifizierten Stelle  
**Format** LOCATE X-Koordinate, Y-Koordinate  
**Beispiel** LOCATE 0, 8

**PRINT**

**Funktion** Anzeige von Daten  
**Format** PRINT [Ausgabeelement] [ [ ; ] ] [Ausgabeelement]  
 \* Ein Ausgabeelement kann eine Ausgabesteuerfunktion, ein Zeichenausdruck oder ein numerischer Ausdruck sein.  
**Beispiele** PRINT "A"  
 PRINT B

**PRINT USING**

**Funktion** Anzeige der Ausgabeelemente gemäß einer Format-Spezifikation  
**Format** PRINT USING-Format-Spezifikation: Ausgabeelement [ [ ; ] ] Ausgabeelement \* [ [ ; ] ]  
**Beispiel** PRINT USING "###"; A [ , ]

**TAB**

**Funktion** Bewegt den Cursor waagrecht zur spezifizierten Stelle auf dem Bildschirm.  
**Format** TAB(Tab-Spezifikation)  
**Beispiel** PRINT TAB (3) ; B

• **Tastat ureingabe****INKEY\$ F**

**Funktion** Holt ein Zeichen vom Tasten-Puffer  
**Format** INKEY\$  
**Beispiel** AS=INKEY\$

**INPUT**

**Funktion** Anforderung Dateneingabe von der Tastatur  
**Format** INPUT [ [Aufforderungszeichen] | ' ' ] Variablenname [, Variablenname] \* [ ; ]  
**Beispiel** INPUT A

• **Felder (Arrays)****OPTION BASE**

**Funktion** Spezifiziert den Mindestwert von Index-Grenzangaben  
**Format** OPTION BASE Mindestwert  
**Beispiel** OPTION BASE 1

**DIM**

**Funktion** Bestimmt ein Feld (d.h. reserviert Speicher dafür)  
**Format** DIM-Feldname (Index-Maximalwert [, Index-Maximalwert [, Index-Maximalwert] ] [, Feldname (Index-Maximalwert [, Index-Maximalwert [, Index-Maximalwert] ] ) ]  
**Beispiel** DIM A (10)

• **Maschinensprache-Unterprogramm-Aufruf**



**CALL**

**Funktion** Aufruf eines Unterprogramms in Maschinensprache  
**Format** CALL-Adresse  
 { [ , [Parameter] ] \*, Parameter }  
 [ , Parameter ]  
**Beispiel** CALL 61440

## • Typ-Erklärung

**CDBL**

**Funktion** Wandelt den Parameterwert in eine reelle Zahl mit doppelter Genauigkeit um.  
**Format** CDBL (Parameter)  
**Beispiel** CDBL (A) A wird in einen Typ mit doppelter Genauigkeit umgewandelt.

**CSNG**

**Funktion** Umwandlung eines Parameterwertes in eine reelle Zahl mit einfacher Genauigkeit  
**Format** CSNG (Parameter)  
**Beispiel** CSNG (I) I wird in einen Typ mit einfacher Genauigkeit umgewandelt.

**DEF DBL**

**Funktion** Definiert die Variablen und vom Benutzer definierte Funktionen, deren Namen mit den spezifizierten Buchstaben beginnen, als reellen Typ mit doppelter Genauigkeit  
**Format** DEF DBL Zeichen [-Zeichen] [, Zeichen [-Zeichen]]\*  
**Beispiel** DEF DBL A-Z Definiert Variablen, deren Namen mit A bis Z beginnen als reellen Typ mit einfacher Genauigkeit.

**DEF SNG**

**Funktion** Definiert Variablen und vom Benutzer definierte Funktionen, deren Namen mit den spezifizierten Buchstaben beginnen als reellen Typ mit einfacher Genauigkeit.  
**Format** DEF SNG Zeichen [-Zeichen] [, Zeichen [-Zeichen]]\*  
**Beispiel** DEF SNG A-Z Definiert Variablen, deren Namen mit A bis Z beginnen als reellen Typ mit einfacher Genauigkeit.

**DEF STR**

**Funktion** Definiert Variablen und vom Benutzer definierte Funktionen, deren Namen mit den spezifizierten Buchstaben beginnen als Zeichen-Typ.  
**Format** DEF STR Zeichen [-Zeichen] [, Zeichen [-Zeichen]]\*  
**Beispiel** DEF STR A-Z Definiert Variablen, deren Namen mit A bis Z beginnen als Zeichen-Typ.

## • Mathematische Funktionen

**ACS** F

**Funktion** Inverse trigonometrische Kosinus-Funktion (arccos oder  $\cos^{-1}$ )  
**Format** ACS (Parameter)  
 Das Ergebnis ist innerhalb des Bereiches:  
 $0^\circ \leq \text{ACS}(x) \leq 180^\circ$   
**Beispiel** ACS (0.5)

**ANGLE**

**Funktion** Spezifiziert die Winkel-Betriebsart für trigonometrische Funktionen  
**Format** ANGLE Winkelspezifikation  
 \* 0 : DEG 1 : RAD 2: GRAD  
**Beispiel** ANGLE 0 Stellt die Winkel-Betriebsart auf DEG ein.

**ASN** F

**Funktion** Inverse trigonometrische Sinus-Funktion (arcsin oder  $\sin^{-1}$ )  
**Format** ASN (Parameter)  
 Das Ergebnis ist innerhalb des Bereiches  
 $-90^\circ \leq \text{ASN}(x) \leq 90^\circ$   
**Beispiel** ASN (1/3)

**ATN** F

**Funktion** Inverse trigonometrische Tangenten-Funktion (arctan oder  $\tan^{-1}$ )  
**Format** ATN (Parameter)  
 Das Ergebnis ist innerhalb des Bereiches  
 $-90^\circ \leq \text{ATN}(x) \leq 90^\circ$   
**Beispiel** ATN

**COS** F

**Funktion** Trigonometrische Kosinus-Funktion ( $\cos x$ )  
**Format** COS (Parameter)  
 \*  $-1440^\circ < \text{Winkel} < 1440^\circ$   
**Beispiel** COS (45)

**EXP** F

**Funktion** Exponentialfunktion ( $e^x$ )  
**Format** EXP (Parameter) \* Parameter < 230  
**Beispiel** EXP (3)

**LGT** F

**Funktion** Zehnerlogarithmus-Funktion ( $\log_{10} x$ )  
**Format** LGT (Parameter) \*  $0 < \text{Parameter}$   
**Beispiel** LGT (3)

**LOG** F

**Funktion** Natürlicher Logarithmus Funktion ( $\log_e x$ ).  
**Format** LOG (Parameter) \*  $0 < \text{Parameter}$   
**Beispiel** LOG (3.6)

**SIN** F  
Funktion Trigonometrische Sinus-Funktion (sin x)  
Format SIN (Parameter) \*  $-1440^\circ < \text{Winkel} < 1440^\circ$

**SQR** F  
Funktion Quadratwurzel ( $\sqrt{x}$ )  
Format SQR (Parameter) \*  $0 < \text{Parameter}$

**TAN** F  
Funktion Trigonometrische Tangenten-Funktion (tan x)  
Format TAN (Parameter) \*  $-1440^\circ < \text{Winkel} < 1440^\circ$

**ABS** F  
Funktion Gibt den absoluten Wert des Parameters (|x|)  
Format ABS (Parameter)  
Beispiel  $X = \text{ABS}(A + B + 3)$

**SGN** F  
Funktion Gibt einen Wert von -1, 0 oder 1 je nach Vorzeichen des Parameters (Signumelement)  
Format SGN (Parameter)  
\* Gibt -1 als negativen Parameter 0 für 0 und 1 für positiv  
Beispiel  $X = \text{SGN}(A + B - 3)$

**INT** F  
Funktion Gibt die größte Ganzzahl, die kleiner oder gleich dem spezifizierten Parameter ist.  
Format INT (Parameter)  
Beispiel  $X = \text{INT}(\text{RND}(1) \times 9 + A)$

**FIX** F  
Funktion Gibt den Ganzzahl-Anteil des Parameters  
Format FIX (Parameter)  
\* Unterscheidet sich von INT nur hinsichtlich der negativen Parameter.  
Beispiel  $\text{FIX}(A + B - 2)$

**FRAC** F  
Funktion Gibt den Wert des Bruch-Anteils eines Parameters  
Format FRAC (Parameter)  
Beispiel  $\text{FRAC}(A/3)$

**ROUND** F  
Funktion Gibt den durch Rundung an der spezifizierten Stelle erhaltenen Wert.  
Format ROUND (Parameter, Stellen-Position)  
Beispiel  $\text{ROUND}(A, -3)$

**RND** F  
Funktion Erzeugung einer Zufallszahl  
Format RND (Parameter)

\* Wenn der Parameter positiv ist, wird aus einer (wiederholbaren) Standardreihenfolge eine Pseudozufallszahl gegeben. Wenn der Parameter 0 ist, wird die gleiche Zufallszahl, die bei der zuverigen Verwendung der Funktion erhalten wurde, gegeben. Wenn der Parameter negativ ist, wird eine echte Zufallszahl (aus einer nicht wiederholbaren Folge) gegeben.

Beispiel  $A = \text{RND}(1)$

**RANDOMIZE**  
Funktion Verändert die Zufallszahlenfolge.  
Format RANDOMIZE  
Beispiel RANDOMIZE

• Zeichendaten-Veränderung

**ASC** F  
Funktion Gibt den ASCII-Code des ersten Zeichens eines Zeichenstrings.  
Format ASC (Zeichenstring)  
Beispiel  $\text{ASC}(A\$)$

**CHR\$** F  
Funktion Produziert das durch den spezifizierten Code dargestellte Zeichen.  
Format CHR\$(Code) \*  $0 \leq \text{Code} < 256$   
Beispiel  $\text{ASC}(A\$)$

**LEFT** F  
Funktion Produziert die spezifizierte Anzahl Zeichen vom linken Ende des Zeichenstrings  
Format LEFT\$(Zeichenstring, Anzahl der Zeichen)  
Beispiel  $A\$ = \text{LEFT}\$(B\$, X)$

**LEN** F  
Funktion Gibt die Länge eines Zeichenstrings an.  
Format LEN (Zeichenstring)  
Beispiel  $A = \text{LEN}(B\$)$

**MID\$** F  
Funktion Produziert einen spezifizierten Substring eines Zeichenstrings  
Format MID\$(Zeichenstring, Position, Anzahl der Zeichen)  
Beispiel  $A\$ = \text{MID}\$(B\$, X-1, Y+1)$

**RIGHT\$** F  
Funktion Gibt die spezifizierte Anzahl der Zeichen vom rechten Ende des spezifizierten Zeichenstrings an.  
Format RIGHT\$(Zeichenstring, Anzahl der Zeichen)  
Beispiel  $A\$ = \text{RIGHT}\$(B\$, X+2)$



**STR\$** F

**Funktion** Wandelt den numerischen Parameter in einen den Parameter darstellenden Zeichenstring um.  
**Format** STR\$ (Parameter)  
**Beispiel** A\$ = STR\$ (B)

**VAL** F

**Funktion** Wandelt den eine Zahl darstellenden String in den entsprechenden numerischen Wert um.  
**Format** VAL (Zeichenstring)  
**Beispiel** B = VAL (A\$)

• **Definitions-Funktionen****DEFFN**

**Funktion** Definiert eine vom Benutzer definierte Funktion  
**Format** DEFFN-Funktionsname [ (Formalparameter [ , Formalparameter ]\* ) ] = Ausdruck  
**Beispiel** DEFFN X(X, Y, Z) = Y+Y\*Z+1

**FN** F

**Funktion** Ansführung einer vom Benutzer definierten Funktion  
**Format** FN-Funktionsname [ (Realparameter [ , Realparameter ]\* ) ]  
**Beispiel** A = FN X(B+1, C, 5)

• **Sonstige Funktionen****FRE** F

**Funktion** Gibt die Größe des verbleibenden unbenutzten Speichers an.  
**Format** FRE [ (Ausdruck) ] \* Die Größe wird in Bytes angegeben.  
**Beispiel** PRINT FRE (1)

**PEEK** F

**Funktion** Gibt den Inhalt einer spezifizierten Speicheradresse an.  
**Format** PEEK (Adresse) \*  $0 \leq \text{Adresse} < 65536$   
**Beispiel** A = PEEK (B)

**POKE** F

**Funktion** Schreibt Daten in eine spezifizierte Speicheradresse  
**Format** POKE Adresse, Daten  
 \*  $0 \leq \text{Adress} < 65536$ ,  $0 \leq \text{Daten} < 256$   
**Beispiel** POKE X, Y+1

**DATES**

**Funktion** Angabe des Datums als String  
**Format** DATES\$  
**Beispiel** PRINT DATES\$ : DATES\$ = 83/11/10"

**TIMES**

**Funktion** Angabe der Zeit als String  
**Format** TIMES\$  
**Beispiel** PRINT TIMES\$ : TIMES\$ = "19:56:25"

• **Statistik-Funktionen****STAT CLEAR**

**Funktion** Initialisiert die Statistik-Funktionen  
**Format** STAT CLEAR  
**Beispiel** START CLEAR

**STAT**

**Funktion** Eingabe der statistischen Daten  
**Format** STAT X Daten [ , Y Daten ]  
**Beispiel** STAT X, Y

**CNT** F

**Funktion** Gibt die Anzahl der statistischen Datenpunkte an.  
**Format** CNT  
**Beispiel** A = CNT

**SUMX** F

**Funktion** Gibt die Summe der X-Daten  
**Format** SUMX  
**Beispiel** B = SUMX

**SUMY** F

**Funktion** Gibt die Summe der Y-Daten  
**Format** SUMY  
**Beispiel** C = SUMY

**SUMX2** F

**Funktion** Gibt die Summe der Quadrate der X-Daten  
**Format** SUMX2  
**Beispiel** B = SUMX2

**SUMY2** F

**Funktion** Gibt die Summe der Quadrate der Y-Daten  
**Format** SUMY2  
**Beispiel** E = SUMY2

**SUMXY** F

**Funktion** Gibt die Summe der Produkte der X-Daten und Y-Daten  
**Format** SUMXY  
**Beispiel** F = SUMXY

**MEANX** F

**Funktion** Gibt den Mittelwert der X-Daten  
**Format** MEANX  
**Beispiel** G = MEANX

**MEANY** F

**Funktion** Gibt den Mittelwert der Y-Daten

Format MEANY  
Beispiel A = MEANY

#### SDX F

Funktion Gibt die Proben-Standardabweichung der X-Daten  
Format SDX  
Beispiel B = SDX

#### SDY F

Funktion Gibt die Proben-Standardabweichung der Y-Daten  
Format SDY  
Beispiel C = SDY

#### SDXN F

Funktion Gibt die Gesamtzahl-Standardabweichung der X-Daten  
Format SDXN  
Beispiel D = SDXN

#### SDYN F

Funktion Gibt die Gesamtzahl-Standardabweichung der Y-Daten  
Format SDXN  
Beispiel E = SDYN

#### LRA F

Funktion Gibt den Ausdruck der linearen Regressionskonstante  
Format LRA  
Beispiel F = LRA

#### LRB F

Funktion Gibt den linearen Regressionskoeffizienten  
Format LRB  
Beispiel G = LRB

#### • Graphik

#### INIT

Funktion Etabliert die Benutzer-Koordinaten  
Format INIT (X-Koordinate, Y-Koordinate), X-Inkrement, Y-Inkrement  
Beispiel INIT (0, 0), 1, 1

#### DRAW/DRAWC

Funktion Zeichnet Punkte und gerade Linien auf der Anzeige (wiederholtes Zeichnen ist ebenfalls möglich)  
Format  $\left\{ \begin{array}{l} \text{DRAW} \\ \text{DRAWC} \end{array} \right\} [ - ] (X, Y) [ [ : ] (X', Y') ] *$   
\*DRAWC löscht die mit der DRAW-Anweisung gezeichneten Punkte und geraden Linien  
Beispiel DRAW (2, 5) - (3, 6), (4, 8)

#### QUAD/QUADC

Funktion Zeichnet ein Rechteck gemäß den spezifizierten Ecken-Koordinaten  
Format  $\left\{ \begin{array}{l} \text{QUAD} \\ \text{QUADC} \end{array} \right\} [(X, Y)] - (X', Y')$   
Beispiel QUAD (2, 5) - (6, 9)  
QUADC (3, 6) - (4, 7)

#### POINT

Funktion Gibt an, ob ein auf der Anzeige spezifizierter Punkt ein- oder ausgeschaltet ist.  
Format POINT (X-Koordinate, Y-Koordinate)  
Beispiel A = POINT (5, 3 + B)

#### • Drucker-Steuerung

#### LLIST

Funktion Druckt ein Programm auf dem Drucker  
Format LLIST [erste Zeilennummer] [ | | ] [letzte Zeilennummer] | ]  
Beispiel LLIST -500 LLIST 100-200 LLIST 100-  
LLIST LLIST 500

#### LPRINT

Funktion Ausgabe der Zeichen auf dem Drucker  
Format LPRINT [Ausgabeelement] [ [ : ] | [Ausgabeelement] ] \*  
\* Ein Ausgabeelement ist z.B.:  
 $\left\{ \begin{array}{l} \text{TAB (numerischer Ausdruck)} \\ \text{Numerischer Ausdruck} \\ \text{Zeichen-Ausdruck} \end{array} \right.$   
Beispiel LPRINT "A ="; A, "B = "; B

#### TAB

Funktion Druckt Leerstellen bis zur spezifizierten Druckstelle.  
Format TAB (Tab-Spezifikation)  
Beispiel LPRINT TAB (A) ; "A"

#### LPRINT USING

Funktion Druckt Ausgabeelemente im spezifizierten Format auf dem Drucker.  
Format LPRINT USING Formatspezifikation; Ausgabeelement[ { ; } ] \*  
Beispiel LPRINT USING "####"; A, B

#### • Dateiverarbeitung

#### CLOSE

Funktion Schließt (und erklärt das Ende der Benutzung von) Dateien.  
Format CLOSE [ [ # ] Dateinummer] [, [ # ] Dateinummer] \*  
Beispiel CLOSE

**MOUNT**

**Funktion** Spezifiziert die Anzahl der Dateipuffer.  
**Format** MOUNT Anzahl der Puffer  
**Beispiel** MOUNT 1

**OPEN**

**Funktion** Öffnet Dateien (erklärt den Anfang der Datei-  
 verarbeitung)  
**Format** OPEN Dateideskriptor [FOR {INPUT | AS[#] |  
 Dateinummer {OUTPUT}]  
**Beispiel** OPEN "CAS0 : TEST" FOR INPUT AS #1

## • Sequentielle Verarbeitung

**EOF**

**Funktion** Anzeige, ob das Ende einer Datei erreicht ist.  
**Format** EOF (Dateinummer)  
**Beispiel** IF EOF (1) = 0 THEN 500

**INPUT#**

**Funktion** Liest Daten von einer sequentiellen Datei  
**Format** INPUT# Dateinummer, Variablenname [, Variablen-  
 name] \*  
**Beispiel** INPUT#1, A

**PRINT#**

**Funktion** Ausgabe von Daten an eine sequentielle Datei  
**Format** PRINT Dateinummer [, Ausgabeelement [ ; [Aus-  
 gabeelement] ] \* ]  
**Beispiel** PRINT #1, AS

**PRINT# ~ USING**

**Funktion** Ausgabe von Daten an eine sequentielle Datei im  
 spezifizierten Format  
**Format** PRINT# Dateinummer, USING-Format Spezifika-  
 tion ; Ausgabeelement [ ; Ausgabeelement ]  
 \* [ { ; } ]  
**Beispiel** PRINT#1, USING "####"; X (1)

## • Direktverarbeitung

**FIELD**

**Funktion** Assoziiert Zeichenvariablenamen mit einem Da-  
 teipuffer  
**Format** FIELD [#] Dateinummer, Länge des Zeichen-  
 strings AS-Zeichenvariablenname [ , Zeichenstring  
 Länge AS-Zeichenvariablenname ] \*  
**Beispiel** FIELD#1, 128 AS HS, 128 AS LS

**GET**

**Funktion** Liest einen Satz in einen Datei-Puffer  
**Format** GET [#] Dateinummer [, Satznummer]  
**Beispiel** GET #1, LP

**PUT**

**Funktion** Schreibt Daten von einem Datei-Puffer in eine  
 Datei  
**Format** PUT [#] Dateinummer [, Satznummer]  
**Beispiel** PUT #1, I

**RSET**

**Funktion** Eingabe von Daten mit rechtem Randausgleich in  
 eine durch die FIELD-Anweisung definierte Variable  
**Format** RSET Zeichenvariablenname = Zeichenausdruck  
**Beispiel** RSET HS = BS

**LSET**

**Funktion** Eingabe von Daten mit linkem Randausgleich in  
 eine durch die FIELD-Anweisung definierte Variable  
**Format** LSET Zeichenvariablenname = Zeichenausdruck  
**Beispiel** LSET LS = "DATA"

**MKSS** F

**Funktion** Wandelt einen numerischen Wert (reeller Typ mit  
 einfacher Genauigkeit) in die entsprechende Zeichen-  
 Darstellung um.  
**Format** MKSS (Parameter)  
**Beispiel** LSET SS = MKSS (SQR (2))

**MKDS** F

**Funktion** Wandelt einen numerischen Wert (reeller Typ mit  
 doppelter Genauigkeit) in die entsprechende Zei-  
 chen-Darstellung um.  
**Format** MKDS (Parameter)  
**Beispiel** RSET DS = MKDS (1)

**CVS** F

**Funktion** Wandelt einen Zeichenstring, der einen numerischen  
 Wert (reeller Typ mit einfacher Genauigkeit) dar-  
 stellt, in den entsprechenden numerischen Wert um.  
**Format** CVS (Parameter)  
**Beispiel** A (1) = CVS (SS)

**CVD** F

**Funktion** Wandelt einen Zeichenstring, der einen numerischen  
 Wert (reeller Typ mit doppelter Genauigkeit) dar-  
 stellt, in den entsprechenden numerischen Wert um.  
**Format** CVD (Parameter)  
**Beispiel** PRINT CVD (DS)

**LOC** F

**Funktion** Gibt bei Verarbeitung mit wahlfreiem Zugriff,  
 die auf den letzten verarbeiteten Satz folgende  
 Satznummer an.  
**Format** LOC (Dateinummer)  
**Beispiel** PRINT USING "(LOC = ####)"; LOC (1)



**LOF** F


**Funktion** Gibt die Anzahl der Sätze in einer Datei an.  
**Format** LOF (Dateinummer)  
**Beispiel** A = LOF (1)

## • Dateiveränderung

**FORMAT**

**Funktion**  
**Format** FORMAT [Datei-Deskriptor]  
**Beispiel** FORMAT 

**FILES**

**Funktion** Anzeige der Namen und Attribute aller Dateien auf einer Diskette  
**Format** FILES [Datei-Deskriptor]  
 \* Dateideskriptor = "0:"  
**Beispiel** FILES 

**KILL**

**Funktion** Löscht eine spezifizierte Datei.  
**Format** KILL [Datei-Deskriptor]  
**Beispiel** KILL "TEST"

## • Programmdateien

**SAVE**

**Funktion** Speichert ein Programm in einer spezifizierten Datei  
**Format** SAVE Datei-Deskriptor [, Paßwort] [, A]  
**Beispiel** SAVE "CAS0 : PROG1" , "KI" , A

**LOAD**

**Funktion** Lädt ein Programm in den Speicher  
**Format** LOAD Datei-Deskriptor [, R ]  
**Beispiel** LOAD "TEST" , R

## • Cassettenrecorder

**VERIFY**

**Funktion** Überprüfung der aus Magnetband gespeicherten Dateien  
**Format** VERIFY Datei-Deskriptor  
**Beispiel** VERIFY "CAS0 : START"

**CETL-Fehlermeldungs-Liste**

Meldung	Fehlerbeschreibung
OF-Fehler	CETL-Dateibereich reicht nicht aus.
OR-Fehler	Eingabeparameter nicht innerhalb des gültigen Bereiches.
NE-Fehler	Nichtexistenter Satzname oder Postenname.
IA-Fehler	Unzulässige Antwort. Die Wahl ist ungültig.

**BASIC Fehlermeldungs-Liste**

Meldung	Fehlerbeschreibung
OB-Fehler	Unzureichender BASIC-Speicherbereich oder Überlauf des System-Stacks aufgetreten.
SN-Fehler	Formatfehler eines Kommandos oder einer Anweisung.
IO-Fehler	Interner Systemfehler, der während der Datenübertragung zwischen einem E/A-Gerät und dem Computer auftrat.
ST-Fehler	Kein Zeichenbereich mehr verfügbar oder die Länge eines Zeichenstrings überschreitet 255 Zeichen.
TC-Fehler	Übermäßig komplexer Ausdruck. In zwei oder mehr einfache Ausdrücke aufteilen.
BV-Fehler	Überlauf des E/A-Puffers.
NR-Fehler	E/A-Gerät nicht zur Eingabe oder Ausgabe bereit.
RW-Fehler	E/A-Gerät kann Daten nicht korrekt schreiben oder lesen.
BN-Fehler	Dateinummer einer nicht geöffneten Datei wurde verwendet.
NF-Fehler	Die spezifizierte Datei wurde nicht gefunden.
TS-Fehler	Eine falsche Spur/Sektor-Nummer wurde spezifiziert.
FL-Fehler	Unternommener Versuch, auf eine Disk mit unzureichendem freien Speicherraum zu schreiben.
OV-Fehler	Das Ergebnis einer Operation oder der Wert von Eingabedaten hat den zulässigen Bereich überschritten.
MA-Fehler	Ein mathematischer Fehler z.B. Division durch 0.
DD-Fehler	Das gleiche Feld oder die gleiche vom Benutzer definierte Funktion wurde zweimal definiert.
BS-Fehler	Ein Index in einem Feld liegt außerhalb des zulässigen Bereiches.

## BASIC Fehlermeldungs-Liste

Meldung	Fehlerbeschreibung
FC-Fehler	Unkorrekter Aufruf einer Funktion oder Anweisung.
UL-Fehler	Eine nicht existente Zeilennummer wurde durch GOTO, GOSUB usw. spezifiziert.
MO-Fehler	Ein erforderlicher Parameter ist nicht in der Anweisung enthalten.
TM-Fehler	Die Variablentypen in einem Ausdruck (linke Seite, rechte Seite, Funktions-Parameter usw.) passen nicht zueinander.
PR-Fehler	Unternommener Versuch, ein Programm oder Daten in eine durch Paßwort oder Schreibschutzschlitz geschützte Datei zu schreiben.
DATA-Fehler	Versuchte READ-Anweisung zum Lesen nicht existenter Daten.
FOR-Fehler	Für die NEXT-Anweisung gibt es keine FOR-Anweisung.
NEXT-Fehler	Für die FOR-Anweisung gibt es keine NEXT-Anweisung.
GOSUB-Fehler	Ein GOSUB ist nicht korrekt mit einem RETURN assoziiert.
FN-Fehler	Undefinierte FN-Funktion.
FORMAT-Fehler	Die Disk wurde nicht formatiert oder ihr Format wurde zerstört.
MOUNT-Fehler	Mehr externe Dateien wurden geöffnet als Puffer vorhanden waren.
FIELD-Fehler	In einer FIELD-Anweisung wurde ein 255 Bytes überschreitender Bereich spezifiziert.

## C85-BASIC SPEZIFIKATION

Verarbeitetes Zeichen		Groß- und Kleinbuchstaben, numerische Zeichen und Spezialsymbole		
Konstanten	Numerisches Zeichen	Einfache Genauigkeit	Anzahl der bedeutungsvollen Stellen	6
		Doppelte Genauigkeit	Anzahl der bedeutungsvollen Stellen	16
		Exponentenbereich	Maximum Minimum	+99 -99
	Zeichen	Max. Länge des Zeichenstrings		255
Variablen	Variablenname			Alphanumerische und Symbole 255
	Zeichen des Variablenattributs	Einfache Genauigkeit	!	
		Doppelte Genauigkeit Zeichenstring	# \$	
	Feldvariable	Max. Anzahl der Elemente		Abhängig von Größe des Benutzerspeichers
		Anzahl der Dimensionen		Bis zu 3
	Max. Dateinamen Länge			8 + 3
Max. Länge der PF-Tastendefinition			15	
Anweisung	Max. Länge der Zeichen pro Zeile		255	
	Höchste Zeilennummer		64999	
Arithmetische Operationen	Addition		+	
	Subtraktion		-	
	Multiplikation		*	
	Division		/	
	Exponentialrechnung		^	
Logische Operationen	Überschuß		MOD	
	Logisches AND		AND	
	Logisches OR		OR	
	Negation		NO	
Exklusives OR		XC XOR		
Verschachtelung FOR-NEXT			(innerhalb der Speicherkapazität)	
Unterprogramm-Verschachtelung			(innerhalb der Speicherkapazität)	
Mehrfachanweisungszeilen			Möglich	
Formatierte Ausgabe			USING-Anweisung	

# Zeichencode-Tabelle

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		(SP)	0	@	P	`	p									≡ X
1	*	(DEL)	!	1	A	Q	a	q								≡ 円
2		(INS)	"	2	B	R	b	r								≡ 年
3		(END)	#	3	C	S	c	s								≡ 月
4		▷	\$	4	D	T	d	t								▷ 日
5		◁	%	5	E	U	e	u								◁ 時
6		△	&	6	F	V	f	v								△ 分
7		▽	'	7	G	W	g	w								▽ 秒
8		(	8	H	X	h	x									♠ 千
9		)	9	I	Y	i	y									♥ 市
A		*	:	J	Z	j	z									♦ 区
B	(HOME)		+	:	K	[	k	]								♣ 町
C	(CLS)	→	.	<	L	\	l									● 村
D	(CR)	←	-	=	M	]	m									○ 人
E	(NUM)	↑	.	>	N	^	n	~								▧
F		↓	/	?	O	_	o	?								▨

\*(\*) 01 Code → KOMMANDO-MENU

In dieser Tabelle nicht aufgeführte Zeichen werden als Leerstellen ausgeschrieben.

Der 7F und der FF-Code können mit dem Drucker nicht gedruckt werden.