Hewlett-Packard Tischrechner 9825A
Bedienung und Programmierung

BEDIENUNGS- UND PROGRAMMIERUNGS-HANDBUCH

Hewlett-Packard Tischrechner 9825A

ZUSAMMENFASSUNG

INHALTSANGABE

ABSCHNITT 1: AUFSTELLEN DES RECHNERS

ABSCHNITT 2: ALLGEMEINES

ABSCHNITT 3: TASTATUR UND PROGRAMMIERUNG

ABSCHNITT 4: DAS TASTENFELD

ABSCHNITT 5: PROGRAMMBEFEHLE, FUNKTIONEN UND OPERATOREN

ABSCHNITT 6: PROGRAMMKORREKTUR

ABSCHNITT 7: ANWEISUNGEN

ABSCHNITT 8: LIVE KEYBOARD

ABSCHNITT 9: MAGNETBANDKASSETTE

ANHANG A: SYNTAX

ANHANG B: FEHLERHINWEISE

ANHANG C: PROGRAMMIER-HINWEISE

ANHANG D: RECHNER-STATUS

ANHANG E: UNTERSCHIEDE ZWISCHEN MODELL 9825A und 9820A/9821A

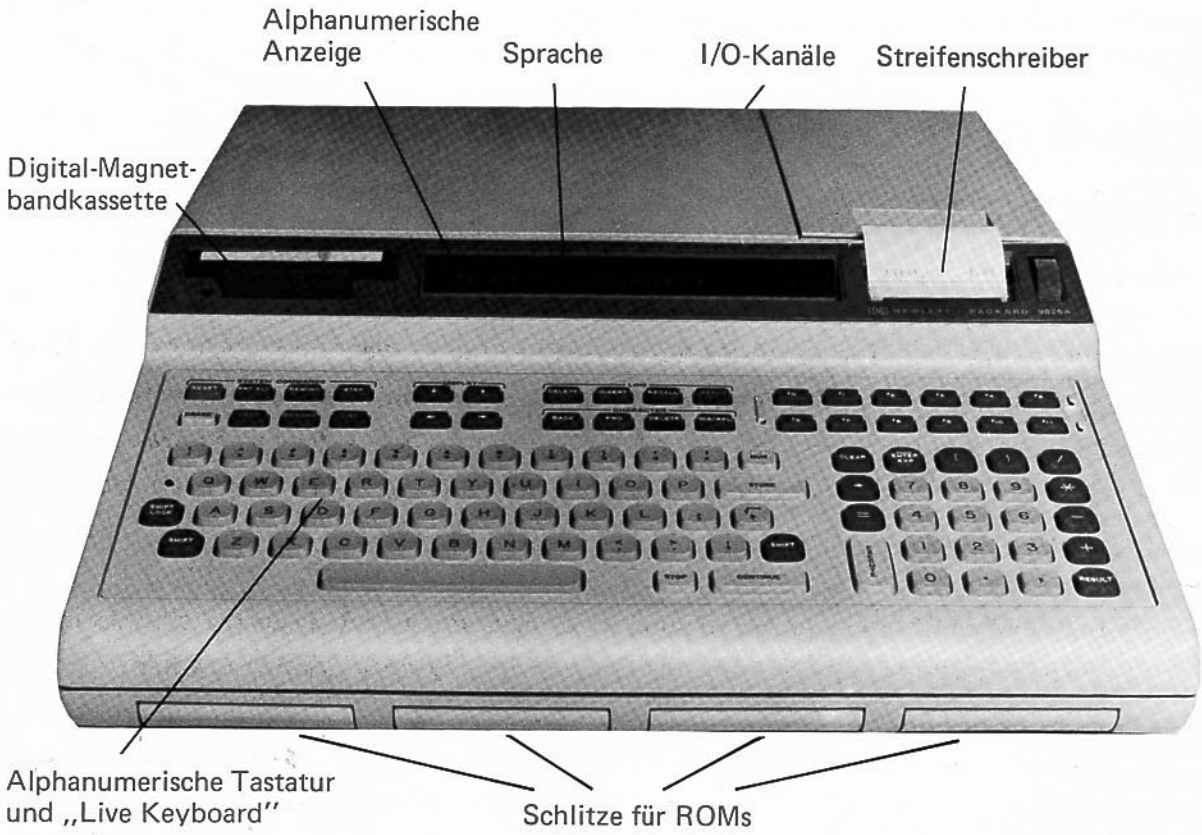
ANHANG F: FEHLERANZEIGE FÜR DIE BANDKASSETTE

ANHANG G: TISCHINSTALLATION

ABBILDUNGEN

TABELLEN

ÜBERSICHT



Alphanumerische Tastatur — mit Groß- und Kleinbuchstaben, wie bei einer Schreibmaschine

Sprache — Abkürzungen, Programmzeilen mit mehreren Befehlen, implizierte Multiplikation

„Live Keyboard" — Auch wenn ein Programm abläuft, steht das Tastenfeld voll zur Verfügung

I/O-Kanäle — Drei Kanäle um Peripherie- oder Meßgeräte anzuschließen. Auf Wunsch zusätzlich mit direktem Speicherzugriff (DMA), Interrupt von Geräten und bit-Verarbeitung

Magnetbandkassette — zwei Spuren mit hoher Schreibdichte gewährleisten schnelle Speicherung von Programmen und Daten

Streifenschreiber — Zeilenbreite bis 16 Zeichen, zum Ausdruck von Programmen, Hinweisen und Daten

Halbleiter-Anzeige — Bis 32 Zeichen in Groß- und Kleinschreibung mit Sonderzeichen zur Anzeige der Eingaben, der Ergebnisse und Hinweise.

ABSCHNITT 1
AUFSTELLEN DES RECHNERS

Bevor Sie Ihren Rechner aufstellen, sollten Sie die folgenden Anweisungen lesen.

Inspektion des Tischrechners

Die einzelnen Geräte Ihres Tischrechner-Systems wurden im Werk sorgfältig überprüft. Überprüfen Sie bitte den Rechner, die ROMs, periphere Geräte und sonstiges mitgeliefertes Zubehör auf Beschädigungen während des Transports. Wenn Sie äußerliche Beschädigungen feststellen, wenden Sie sich zunächst an den Spediteur und benachrichtigen Sie bitte die nächste HP Niederlassung.

Bitte überprüfen Sie auch, ob die Sendung vollständig ist und daß Sie die bestellte Option bekommen haben. Alles mitgelieferte Zubehör ist in der folgenden Tabelle aufgeführt.

Sollten bei Ihrem Rechner Schwierigkeiten auftreten oder wenn Zubehör fehlt, wenden Sie sich bitte an unser nächstes HP Büro. Die Adressen finden Sie auf der Rückseite dieses Handbuches.

Mitgeliefertes Zubehör

Mit jedem Tischrechner HP 9825A wird das folgende Zubehör ausgeliefert. Peripheriegeräte und Interface-Karten werden separat verpackt, wobei die Handbücher oder Bedienungshinweise den jeweiligen Sendungen beigepackt sind.

Mitgeliefertes Zubehör

Beschreibung	Anzahl	Teile-Nummer
Bedienungs- und Programmierhandbuch	1	09825–90095
Kurzformanleitung	2	5952–9022
Fehlerverzeichnis (Handbuch)	1	09825–90015
Leere Magnetbandkassette	1	9162–0061
Programmsammlung	1	09825–10000
Netzkabel	1	8120–1689
Abdeckhaube	1	9222–0495
Magnetkopfreiniger	1	8500–1251
Schablonen für die frei definierbaren Funktionstasten	5	7120–4802
Ersatzsicherung (1A)	1	2110–0001
Ersatzsicherung (2A)	1	2110–0002
Prüf-Magnetbandkassette	1	09825–90035
Handbuch mit Prüfanweisungen	1	09825-90031
Druckerpapier	3	siehe unten*
Programmsammlung	1	9828–0563

Das gesamte Zubehör kann unter der HP-Bestellnummer 09825–80000 nachbestellt werden.

* Druckerpapier wird in Paketen mit 6 Rollen geliefert, die HP Bestellnummer ist 9270–0479.

In Deutschland wird der Tischrechner mit einem Netzkabel mit Schuko-Stecker ausgeliefert. Sollten Sie ein anderes Kabel erhalten haben, so wenden Sie sich bitte an unser nächstes HP-Büro.

Netzanschluß

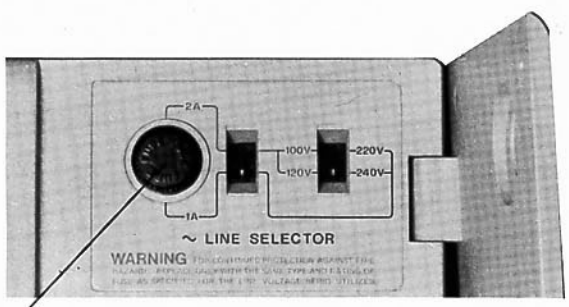
• Netzspannung	100 V (+5 %, −10 %) 120 V (+5 %, −10 %) 220 V (+5 %, −10 %) 240 V (+5 %, −10 %)	Spannung mit Schalter einstellbar
• Netzfrequenz:	48 bis 66 Hz	
• Leistungsaufnahme:	100 V 1,7A 120 V 1,5A 220 V 0,8A 240 V 0,75A	
• Erdung:	Gerät nur an Steckdosen mit Schutzkontakt anschließen.	

Auswechseln der Sicherungen

Für 100 und 120 Volt wird eine 2A-Sicherung eingesetzt; für 200 und 220 Volt eine 1A-Sicherung.

Warnung

Vor dem Einsetzen oder Auswechseln einer Sicherung muß unbedingt das Netzkabel aus der Steckdose gezogen werden.



Sicherung

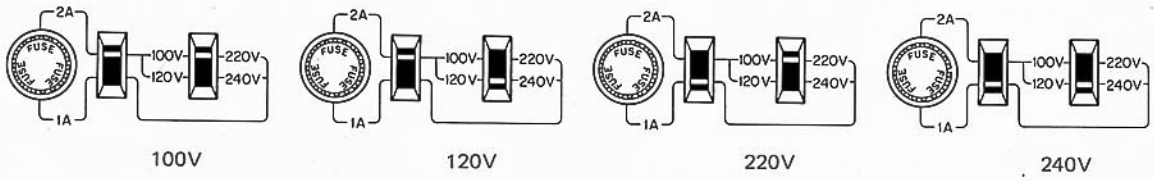
Die Sicherung

Die Abbildung zeigt, wo die Sicherung unter der Abdeckung für die Papierrolle liegt. Um die Sicherung auszuwechseln, wird zuerst das Netzkabel abgezogen. Die Sicherung wird herausgenommen, indem die schwarze Kappe eingedrückt und nach links gedreht wird. Die Sicherung kann dann leicht aus der schwarzen Kappe herausgezogen und ausgewechselt werden. Die schwarze Kappe mit der Sicherung wird dann wieder eingesetzt, angedrückt und nach rechts gedreht bis die schwarze Kappe verriegelt ist.

Einschalten des Gerätes

1. Bevor der Tischrechner an das Netz angeschlossen wird, ist erst zu überprüfen, ob die richtige Sicherung eingesetzt ist.

2. Überprüfen Sie, ob die zwei Spannungswahlschalter auf die richtige Netzspannung eingestellt sind. Die richtige Einstellung können Sie den folgenden Zeichnungen entnehmen. Die Schalter können verschoben werden, indem ein kleiner Schraubenzieher in den Schlitz eingeführt, und der Schalter dann verschoben wird.



Einstellen der richtigen Netzspannung

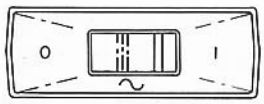
3. Der Systemeinschub auf der rechten Seite des Rechners muß richtig eingeschoben sein und darf nicht vorstehen.

4. Die ROMs und Interfacekarten werden, wie in den jeweiligen Handbüchern beschrieben, eingesetzt.

Warnung

Der Rechner muß immer ausgeschaltet sein, wenn ROMs oder Interfacekarten eingesetzt oder herausgenommen werden.

5. Den Rechner an das Netz anschließen und mit dem Schalter an der rechten Seite einschalten.



Überprüfen des Rechners

Sollte Ihr Rechner nicht ordnungsgemäß arbeiten, so lesen Sie bitte die System-Testanweisung.

Einlegen des Druckerpapiers

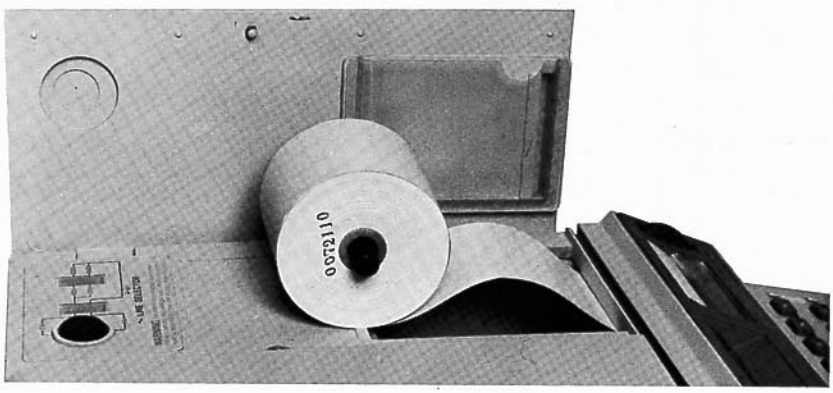
Für den eingebauten Drucker wird wärmeempfindliches Papier mit der HP-Bestell-Nr. 9270—0479 verwendet.

Die Papierrolle wird wie folgt eingelegt:

1. Die Abdeckklappe hochklappen und die leere Spule herausnehmen. Papierreste aus dem Drucker entfernen.

2. Die neue Rolle, wie in folgender Abbildung gezeigt, einsetzen.

3. Das Papier in den Drucker einschieben und mit dem Rändel-Rad weitertransportieren.



Einsetzen des Druckerpapiers

ROMs, Peripheriegeräte und Interfaces

In diesem Abschnitt werden die ROMs, die Peripheriegeräte und die Interfacekarten für den Tischrechner 9825A beschrieben.

ROMs (Festwertspeicher)

Mit jedem ROM wird der Befehlsvorrat des Rechners erweitert. Für den Rechner stehen mehrere ROMs zur Verfügung, die spezielle Aufgaben übernehmen können, wie z. B. die Steuerung von Plottern und weiteren Peripheriegeräten, zusätzlich lassen sich mit ihnen die Programmiermöglichkeiten des Rechners erweitern.

Einsetzen der ROMs

Die ROMs können, wie in der nächsten Abbildung gezeigt, in die Schlitze an der Vorderseite des Rechners eingesetzt werden.

Schlitze zum Einsetzen der ROMs

Bevor ein ROM eingesteckt wird, muß der Rechner abgeschaltet werden. Das ROM wird dann in den Schlitz eingeschoben und so weit eingedrückt, daß es nicht mehr vorsteht.

Die folgenden ROMs stehen in folgenden Kombinationen zur Verfügung.

Zeichenketten-ROM

Mit diesem ROM kann der Rechner Buchstaben und Worte (Zeichenketten) genauso verarbeiten wie sonst Zahlen. Einige der Möglichkeiten mit diesem ROM sind: Einfache Zeichenketten und Zeichenkettenfelder, numerische Werte können aus einer Zeichenkette aussortiert werden, Verkettungen, Anzeige und Ausdruck aller Sonderzeichen sowie die Zuordnung von numerischen Werten in Zeichenketten.

ROM für erweiterte Programmierung

Mit diesem ROM können die Programmiermöglichkeiten des 9825A erweitert werden. Möglichkeiten sind: for next – Schleifen, die Speicherung von Split- und Integerzahlen, Funktionen und Unterprogramme mit mehreren Parametern und cross-reference-Befehlen.

Matrix-ROM

Mit dem Matrix ROM wird der Befehlsvorrat zur Verarbeitung von Matrizen und Feldern erweitert. Es ermöglicht die Bestimmung der Determinanten, die Addition, Subtraktion, Multiplikation, Division, Transponierung und Inversion von Matrizen sowie deren Initialisierung.

Plotter-ROM (für 9862A)

Dieses ROM wird in den Rechner eingesetzt, wenn der Plotter 9862A an den Rechner angeschlossen wird. Es lassen sich Achsen zeichnen und beschriften und Funktionen plotten. In der Betriebsart „Schreibmaschine" können Buchstaben verschiedener Größe geschrieben werden. Es können auch mehrere Plotter 9862A gleichzeitig an den Rechner angeschlossen werden.

Universal I/O-ROM

Mit diesem I/O-ROM lassen sich formatierte Ein- und Ausgaben steuern. Fast alle Peripheriegeräte der Serie 9800, außer dem Plotter, können mit diesem ROM gesteuert werden. Auch kann der Status von Geräten überprüft und der HP-Interface Bus (HP-IB) gesteuert werden.

Erweitertes I/O-ROM

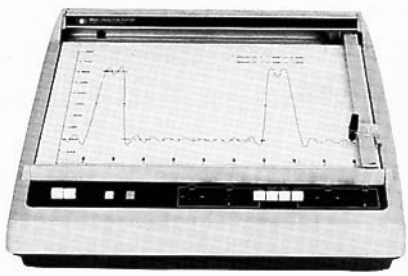
Dieses ROM dient zur vollständigen Steuerung des HP-IB. Interrupt-Befehle, Auto-Start, Fehlerermittlung, Zeitsteuerung, gepufferte Ein/Ausgabe, Bitverarbeitung, Codeumwandlung, Burst Read and Write und direkter Speicherzugriff (DMA) sind möglich.

Peripheriegeräte

Jedes der folgenden Peripheriegeräte wird komplett mit dem erforderlichen Interface und den Bedienungshandbüchern geliefert. Zur Steuerung dieser Peripheriegeräte, außer dem 9862A, ist das Universal I/O-ROM erforderlich.

X-Y Plotter 9862A

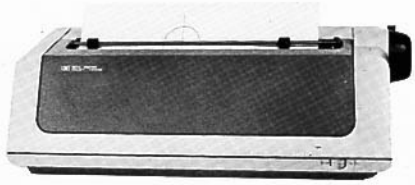
Dieser Plotter zeichnet Histogramme, Schaltbilder, lineare und Log-Log-Darstellungen sowie Polar-Diagramme. Er zeichnet Achsen und beschriftet sie entsprechend. Das Plotter-ROM enthält alle erforderlichen Befehle, um diesen Plotter mit dem Rechner 9825A zu steuern.



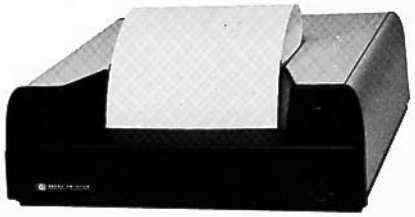
Drucker

Wenn Sie Ihre Ergebnisse in Tabellen, Aufstellungen oder Formblätter eintragen wollen, so sollten Sie einen der folgenden Drucker verwenden.

Der Drucker 9871A hat eine Zeilenbreite von 132 Zeichen, kann mehrere Kopien liefern und hat einen Zeichenvorrat von 96 Zeichen. Die mittlere Druckgeschwindigkeit beträgt 30 Zeichen/Sek. Ein weiterer Vorteil dieses Geräts ist, daß es sich auch als Plotter verwenden läßt.



Der schnelle Thermo-Drucker 9866B kann bis zu 250 Zeilen mit je 80 Anschlägen pro Minute ausdrucken. Insgesamt lassen sich 96 Zeichen mit Groß- und Kleinbuchstaben ausdrucken. Über den Rechner gesteuert, kann der Ausdruck voll formatiert werden. Es lassen sich Tabellen und Zeichnungen erstellen.

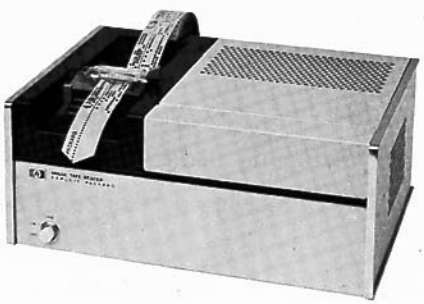


Kartenleser

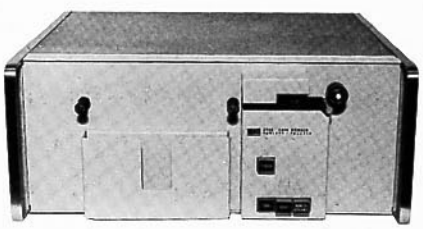
Bei industriellen Anwendungen oder in Schulen ist der Kartenleser 9869A zur Eingabe größerer Datenmengen in Form gelochter oder markierter Karten geeignet. Es lassen sich bis zu 300 Karten mit 80 Spalten pro Minute einlesen.

Lochstreifenleser

Lochstreifen von analytischen Meßgeräten, NC-Maschinen oder Computer-Terminals lassen sich mit einem dieser Lochstreifenleser unmittelbar in den Rechner eingeben. Mit Modell 9863A können beliebige Codes mit Geschwindigkeiten bis 20 Zeichen/Sek. eingelesen werden.



Der für Dauerbetrieb ausgelegte schnelle Streifenleser 9883A liest bis 500 Zeichen pro Sekunde optisch ein.

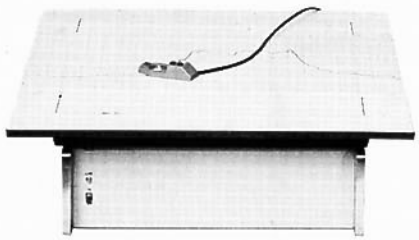


Lochstreifenstanze 9884A

Diese zuverlässige kompakte Stanze erstellt Lochstreifen mit einer Geschwindigkeit bis 75 Zeichen pro Sekunde.

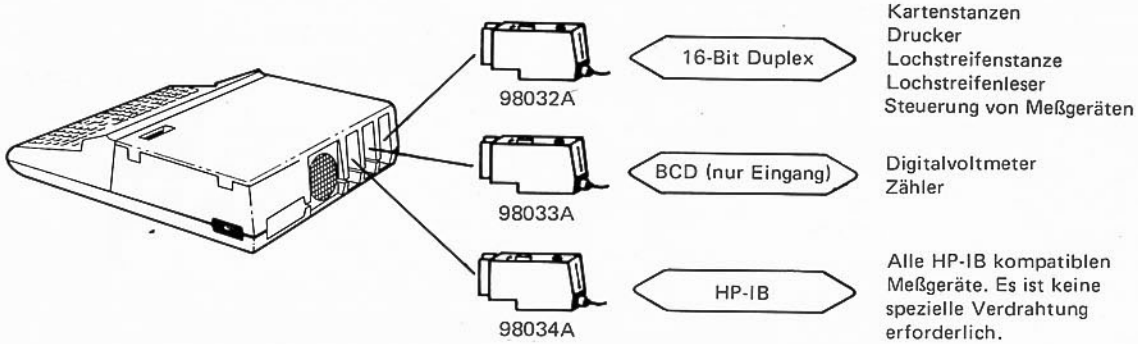
Digitalisierer 9864A

Mit diesem Gerät lassen sich Kurven oder analoge Werte von Zeichnungen als eine Serie diskreter Punkte als digitale x-y-Werte unmittelbar in den Rechner eingeben. Zur Eingabe wird die Kurve mit einem Fadenkreuz nachgefahren. Der Rechner kann dann die eingegebenen Werte auswerten. Mit entsprechenden Programmen können graphische Daten oder Schriebe unmittelbar berechnet werden.



Interfaces

Das folgende Interface steht zum Anschluß von Peripheriegeräten zur Ein- und Ausgabe von Werten zur Verfügung. Jedes Interface, das mit ausführlichen Handbüchern geliefert wird, ermöglicht die Steuerung der Ein- und Ausgabe über den Rechner. Hier einige typische Anwendungen:



16-Bit Duplex Interface

Die Karte 98032A ist eine 16-Bit parallel-zeichenserielle Interface-Karte. Sie erlaubt vollen Duplex-Betrieb zwischen Rechner und angeschlossenem Peripheriegerät. Daten können also gleichzeitig eingelesen und ausgegeben werden. Zur Steuerung dieser Interfacekarte ist das Universal I/O-ROM erforderlich.

BCD Interface

Die BCD-Interfacekarte 98033A erlaubt die Abfrage von Geräten mit Parallel-BCD-Ausgang. Bis zu 10 Stellen, die Mantisse, das Vorzeichen, das Vorzeichen des Exponenten und Überlauf lassen sich eingeben. Zur Steuerung dieser Interface-Karte ist das Universal-I/O-ROM erforderlich.

HP-Interface Bus

Über die HP-IB — Interfacekarte 98034A lassen sich bis zu 14 HP-IB kompatible Geräte über den Bus an den Rechner anschließen. Die Grundfunktionen der Karte werden durch das Universal-I/O-ROM gesteuert. Ist eine vollständige Steuerung erforderlich, muß des erweiterte I/O-ROM in den Rechner eingesetzt werden.

Für weitere Informationen über ROMs, Peripheriegeräte und Interface-Karten wenden Sie sich bitte an unser nächstes HP-Verkaufsbüro.

Programme auf Magnetbandkassetten

Auf Magnetbandkassetten registrierte Programme für viele Arbeitsgebiete stehen fertig zur Verfügung. Eine Kassette mit fertigen Programmen wird mit jedem Rechner mitgeliefert. Wegen einer kompletten Liste der Programme und wegen der Preise wenden Sie sich bitte an unser nächstes Hewlett-Packard-Büro.

Hauszeitschrift „Keyboard"

Unsere Hauszeitschrift „Keyboard" enthält allgemeine Beiträge über Hewlett-Packard-Tischrechner, Peripheriegeräte und Programme. Sie enthält weiterhin von Kunden geschriebene Rechnerprogramme, Beschreibungen der neuesten Geräte und der neuentwickelten Programme, Programmierhinweise sowie Beiträge, die jeden Tischrechner-Benutzer interessieren.

Um diese kostenlose Hauszeitschrift „Keyboard" zu erhalten, brauchen Sie nur die jedem Rechner beigelegte Bestellkarte auszufüllen.

Kundendienst-Wartungsverträge

Hewlett-Packard bietet Ihnen die Möglichkeit, Kundendienst-Wartungsverträge abzuschließen. Solch ein Vertrag umfaßt die vollständige Betreuung und Instandhaltung Ihrer Geräte und Systeme. Dadurch wird die Einsatzbereitschaft Ihrer Tischrechner-Systeme gesteigert.

Wartung, die alle Kosten beinhaltet

Mit dieser Vertragsform sind Sie gegen alle Eventualitäten abgesichert, denn alle Ersatzteile sowie Reisezeiten und Arbeitszeiten sind im Grundpreis enthalten, auch alle anfallenden Reparaturen außerhalb der normalen Wartungsintervalle.

Ein Wartungsvertrag hilft bei der Kostenplanung

Wir bieten unsere Wartungsverträge zu Festpreisen an. Im Falle eines Voll-Wartungsvertrages entstehen Ihnen damit keine weiteren Kosten; so werden unvorhergesehene Belastungen Ihres Budgets ausgeschlossen.
Wartungsverträge von Hewlett-Packard können auf Ihre individuellen Bedürfnisse abgestimmt werden. Unsere technischen Büros beraten Sie gerne.

ABSCHNITT 2
ALLGEMEINES

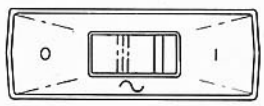
Dieser Abschnitt enthält allgemeine Informationen über den Tischrechner 9825A. Die Einzelheiten werden in den folgenden Abschnitten beschrieben.

Bevor Sie mit dem Rechner arbeiten

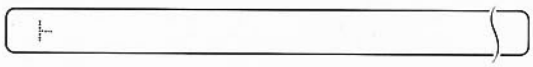
Jedesmal, wenn Sie mit dem Rechner arbeiten wollen, sollten Sie folgendes überprüfen:

Ist der Rechner ausgeschaltet:

- den Netzschalter auf der rechten Seite in Stellung „1" schalten



- erscheint die folgende Anzeige, können Sie mit dem Rechner arbeiten.

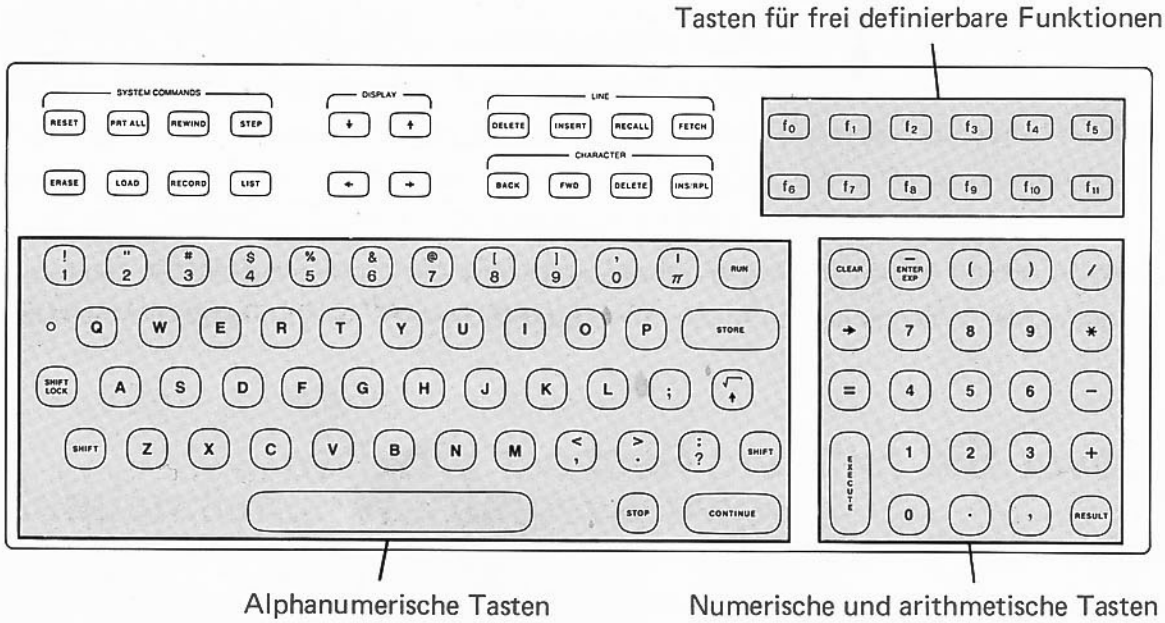


Bleibt die Anzeige dunkel,

- Drücken: (CLEAR) oder (STOP)

Erfolgt noch immer keine Anzeige, überprüfen Sie bitte den Netzanschluß und die Sicherung (Seite 5 und 6). Erscheint dieses Zeichen (⊢), ist der Rechner betriebsbereit.

Beschreibung des Rechners 9825A



Tastenfeld des Modell 9825A

Tastenfeld

Das Tastenfeld des 9825A ist in der obigen Abbildung dargestellt. Es ist in folgende Funktionsbereiche eingeteilt:

- Alphanumerische Tasten – entsprechen etwa der Schreibmaschinentastatur. Um z. B. ein großes A anzuzeigen, werden die Umschalttaste (SHIFT) und die Taste (A) gleichzeitig gedrückt. Zur Anzeige von Prozent werden die Umschalttaste und (%/5) gleichzeitig gedrückt.

- Numerische Tasten – neben den Tasten für die Grundrechenarten enthält dieser Block alle Tasten zur Eingabe von Zahlen. Es können jedoch auch die Zifferntasten der Schreibmaschinentastatur verwendet werden.

- Arithmetische Tasten – neben den Zifferntasten sind die Tasten für die vier Grundrechenarten angeordnet. Die Tasten für die Exponentialfunktionen und Quadratwurzeln (↑) befinden sich auf der rechten Seite der Schreibmaschinentastatur.

- Tasten für frei programmierbare Funktionen – diese mit (f0) bis (f11) bezeichneten Tasten befinden sich in der rechten oberen Ecke. Ihre Verwendung wird in Abschnitt 4 beschrieben.

- Die übrigen Tasten dienen zum Aufrufen weiterer Funktionen, zur Programmkorrektur und zur Steuerung der Anzeige. Auch sie werden ausführlich in Abschnitt 4 beschrieben.

Tasten gleicher Farbe haben ähnliche Funktionen. Wenn Sie sich mit dem Tischrechner 9825A eingearbeitet haben, werden Sie diesen Vorteil erkennen.

Hier noch einige Besonderheiten bei der Eingabe über das Tastenfeld:

- Leertaste — im allgemeinen hat sie keine Bedeutung. Die folgenden Beispiele können Sie eingeben als:

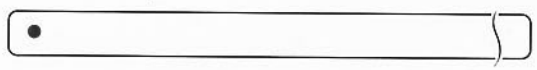
  A+B oder A + B.

  Der Ausdruck wird beide Male gleich behandelt. Die Leertaste ist jedoch bei Labels (Zeichen in Anführungszeichen) und für die Ausgabe z. B. auf dem Drucker und in der Anzeige von Bedeutung.

- Tasten-Wiederholfunktionen — wird eine Taste länger gedrückt, wird ihre Funktion so lange wiederholt, bis sie losgelassen wird. Dies ist speziell bei den Korrekturtasten von Vorteil.

- Das Zeichen ⊢ — ist die Anzeige gelöscht und zur Eingabe bereit, erscheint das Zeichen ⊢ auf der linken Seite. Mit diesem Zeichen wird auch das Ende einer gespeicherten Zeile gekennzeichnet.

- Das Arbeitszeichen — ein kleines rotes Lämpchen in der linken Ecke der Anzeige leuchtet auf, wenn ein Programm abgearbeitet wird.



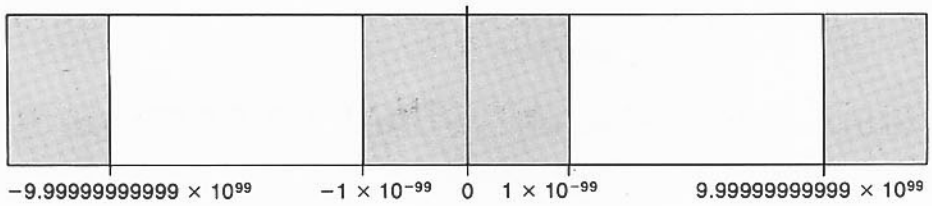
Anzeige und Zeilenlänge

Der Tischrechner 9825A verfügt über eine 32 stellige alphanumerische Anzeige. Obwohl nur 32 Zeichen angezeigt werden können, lassen sich bis 80 Zeichen je Zeile eingegeben. Werden nach dem 32. Zeichen weitere Zeichen eingegeben, so verschiebt sich die gesamte Anzeige nach links. Nach Eingabe von 67 Zeichen ertönt ein Signal als Hinweis, daß nur noch 13 weitere Zeichen eingegeben werden können.

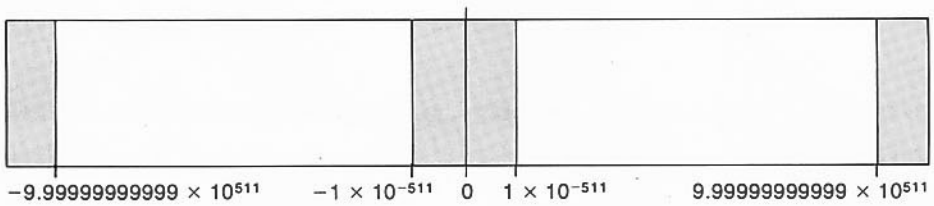
Rechenbereich

Der Bereich der Werte für die Eingabe und die Speicherung reicht von $-9{,}99999999999 \times 10^{99}$ bis -1×10^{-99}, 0, 1×10^{-99} bis $9{,}99999999999 \times 10^{99}$. Es lassen sich jedoch Berechnungen im Bereich von $-9{,}99999999999 \times 10^{511}$, bis -1×10^{-511}, 0 und 1×10^{-511} bis $9{,}99999999999 \times 10^{511}$ ausführen.

Speicherbereich



Rechenbereich



außerhalb des Bereichs  innerhalb des Bereichs 

Dadurch lassen sich Rechnungen durchführen, deren Zwischenergebnisse größer als der Eingabe- und Speicherbereich sind, deren Endergebnisse aber innerhalb des Speicherbereichs liegen. Zum Beispiel:

$$(9{,}2 \times 10^{23} \times 8{,}6 \times 10^{80}) / (1 \times 10^{24})$$

Bei der Multiplikation der beiden ersten Werte ist das Ergebnis:

$$(7912 \times 10^{104})$$

Dieses Zwischenergebnis kann nicht gespeichert werden, das Endergebnis 7912×10^{80} wird jedoch gespeichert.

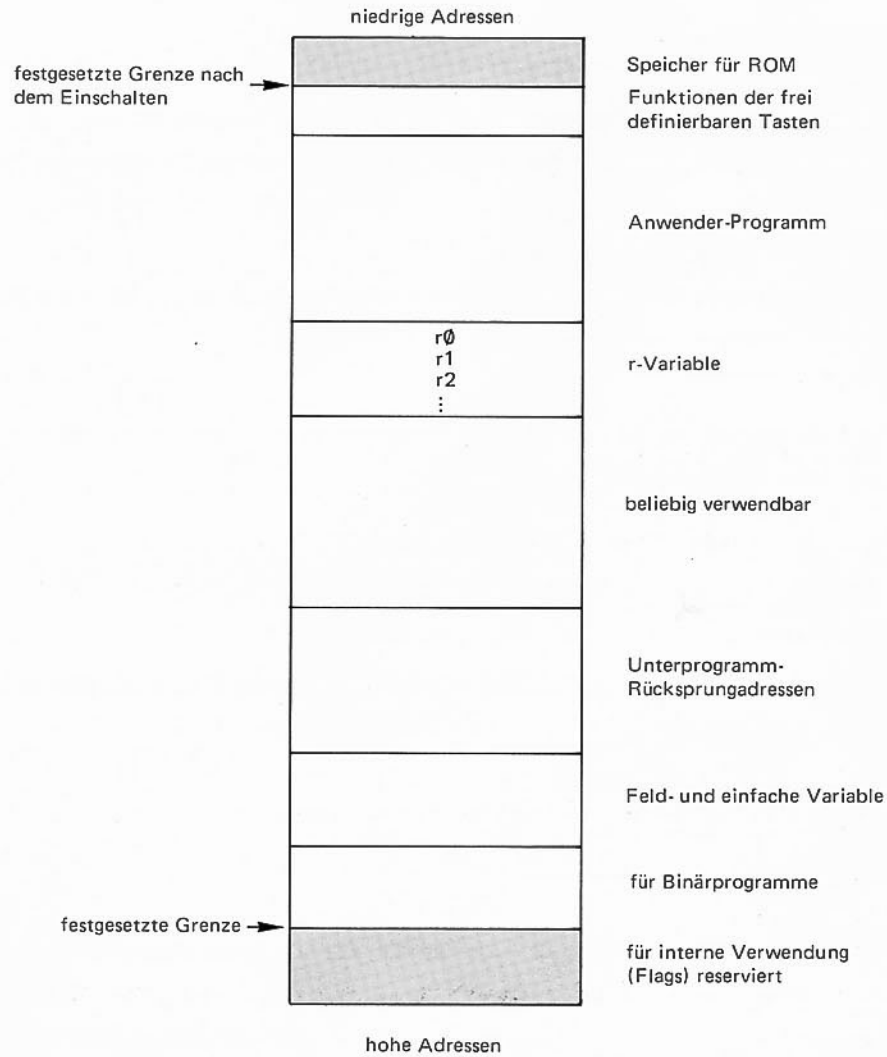
Speicher

Der 9825A besitzt zwei verschiedene Arten Speicher: Einen Read/Write-Speicher und fest verdrahtete Speicher. Im Read/Write-Speicher werden Programme und Daten gespeichert, dabei wird in den Speicher „hineingeschrieben" (Write). Alle so gespeicherten Informationen lassen sich wieder aufrufen oder abarbeiten, also „herauslesen" (Read).

Die fest verdrahteten Speicher enthalten permanente Informationen, die beim Ausschalten des Geräts im Gegensatz zum Read/Write-Speicher nicht gelöscht werden. Fest verdrahtete Speicher, ROMs, lassen sich zusätzlich in das Gerät einsetzen, um den Befehlsvorrat zu erweitern.

Programme und Daten im Read/Write-Speicher können auf Magnetbandkassetten zur späteren Verwendung aufgezeichnet werden.

Aufbau des Read/Write Speichers



Programmiersprache

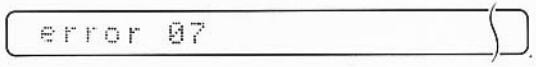
Der Tischrechner 9825A arbeitet mit der Programmiersprache HPL (HP Language). Diese Sprache besteht aus einzelnen kurzen Befehlen. Dies sind Abkürzungen aus der englischen Sprache in Kleinschreibung wie z. B. `prt` für print = ausdrucken. Eine Programmzeile kann mehrere Befehle enthalten, die durch Semikolons getrennt werden.

Zwei weitere Besonderheiten von HPL sind die implizierte Multiplikation und der Zuweisungsoperator. Bei der implizierten Multiplikation braucht das Malzeichen z. B. bei 5x nicht eingegeben zu werden, wodurch die Verarbeitungsgeschwindigkeit erhöht und Speicherkapazität gespart wird. Mit dem Zuweisungsoperator ÷ werden Variablen Werte zugewiesen, wie z. B. `5 ÷ D`.

Der Befehlsvorrat kann durch zusätzliche ROMs jederzeit erweitert werden.

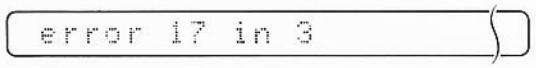
Fehlerhinweise

Bei einem Bedienungs- oder Programmierfehler erfolgt ein akustisches Signal und das Wort „Error" mit einer Zahl erscheint in der Anzeige. Mit der Zahl wird die Art des Fehlers angegeben.



zeigt einen Syntax-Fehler an

Erfolgt eine Fehlermeldung bei Ablauf eines Programms, wird auch die Nummer der Programmzeile, in der der Fehler auftritt, angezeigt.



zeigt an, daß der Parameterbereich in Zeile 3 überschritten ist.

Eine vollständige Liste der Fehlerhinweise enthält der Anhang B und das kleine Heftchen unter der Abdeckung des Rechners.

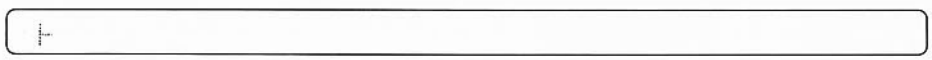
ABSCHNITT 3
TASTATUR UND PROGRAMMIERUNG

Dieser Abschnitt erklärt die Bedienung des Tastenfeldes und die Programmierung. Fast alle über das Tastenfeld ausführbaren Operationen sind auch programmierbar. Die einzelnen Befehle werden noch ausführlich beschrieben.

Systemtasten

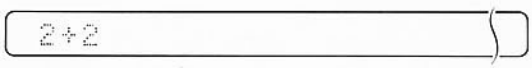
Die folgenden vier Tasten werden oft bei der Bedienung über das Tastenfeld und bei der Programmierung verwendet.

(CLEAR) Löscht die Anzeige, mit dem Zeichen ⊢ wird angezeigt, daß der Rechner zur Eingabe bereit ist.

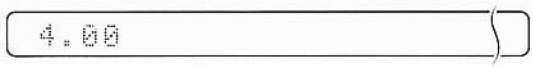


(RUN/CONTIN) Hiermit wird die in der Anzeige stehende Aufgabe ausgeführt. Beispiel: 2 + 2:

Drücken: (2) (+) (2)



Drücken: (RUN/CONTIN)



(STORE) Speichert eine Programmzeile im Speicher.

Beispiel:

Drücken: (7) (→) (SHIFT)* (A)

Drücken: (STORE)

Damit wird diese Programmzeile, die der Variablen A den Wert 7 zuordnet, gespeichert.

*Die (SHIFT) Taste schaltet die folgenden Tasten um.

[RUN] Durch Druck auf diese Taste wird das gespeicherte Programm ab Zeile 0 ausgeführt. Im vorigen Programm wird durch [RUN] die Zahl 7 der Variablen A zugeordnet.

Tastenfeld-Arithmetik

Die sechs grundlegenden arithmetischen Funktionen des Rechners sind: Addition (+), Subtraktion (–), Multiplikation (*), Division (:), Potenzieren (↑) und Quadratwurzel ziehen (√).

Um eine Rechnung, wie z. B. 8 x 2 auszuführen, wird wie folgt eingegeben:

[8] [*] [2]

```
8*2
```

Dann drücken [EXECUTE]

```
16.00
```

Um eine Zahl zu potenzieren, z. B. 8^2, werden gedrückt:

[8] [↑] [2] [EXECUTE]

```
64.00
```

Dabei ist zu beachten, daß 8^{-2} nicht als 8↑ – 2 eingegeben werden kann. Das – 2 muß dabei in Klammern stehen: 8↑(– 2).

Der nach Drücken der Execute - Taste angezeigte Wert wird in dem Ergebnis Speicher (RESULT) gespeichert. Mit der Taste [RESULT] kann dieser Wert in weiteren Rechnungen verwendet werden. Zum Beispiel:

[2] [*] [3] [EXECUTE]

```
6.00
```

[RESULT] [*] [2] [EXECUTE]

```
12.00
```

[RESULT] [/] [4] [EXECUTE]

```
3.00
```

Wenn Sie z. B. folgende Aufgabe eingeben:

```
6000000*9000000
```

so zeigt der Rechner das Ergebnis in Gleitkommadarstellung mit neun Stellen hinter dem Komma an:

```
 5.400000000e 13
```

Der Rechner schaltet automatisch um, da die Zeile für die Anzeige mit zwei Stellen hinter dem Komma zu groß ist, auf die der Rechner nach dem Einschalten automatisch geschaltet ist.

Hierarchie arithmetischer Operationen

Enthält eine Aufgabe mehr als eine arithmetische Operation, so erfolgt die Berechnung nach der folgenden Hierarchie:

√	Radizieren	zuerst ausgeführt
↑	Potenzieren	↓
Kein Operator	Implizierte Multiplikation	
*/	Multiplikation , Division	
+-	Addition, Subtraktion	zuletzt ausgeführt

Diese Hierarchie entspricht der normalen, in der Mathematik verwendeten Folge.

Enthält ein Ausdruck zwei Operationen gleichen Ranges, so werden diese Operationen von links nach rechts ausgeführt.

Klammern ändern die Reihenfolge der Berechnung. Ausdrücke in Klammern werden zuerst berechnet und dann erst die Operationen außerhalb der Klammern. Sind Klammern geschachtelt, wird zuerst die innerste Klammer berechnet. (Abarbeitung von innen nach außen)

Das folgende Beispiel zeigt, wie der Rechner die folgende Aufgabe Schritt für Schritt berechnet. Wenn man weiß, in welcher Reihenfolge das Gerät Berechnungen ausführt, lassen sich auch sehr komplexe Ausdrücke eingeben. Beispiel:

2+3*6↑2(4)/(6-2)↑2	Potenzierung
2+3*36(4)/(6-2)↑2	implizierte Multiplikation
2+3*144/(6-2)↑2	Multiplikation
2+432/(6-2)↑2	Auflösung der Klammer
2+432/4↑2	Potenzierung
2+432/16	Division
2+27	Addition
29	Ergebnis

Variable

Oft ist es erforderlich, numerischen Werten Buchstaben zuzuordnen und mit diesen Buchstaben zu rechnen. Innerhalb eines Programmes können den Buchstaben laufend neue Werte zugeordnet werden – daher der Name Variable. Manchmal wird ein Buchstabe (Variable) zur Zwischenspeicherung verwendet, um z. B. die Summierung einer Reihe von Zahlen zu ermöglichen. Auch können einer Variablen innerhalb einer Rechnung verschiedene Werte zugeordnet werden, um zu sehen, welchen Einfluß dieser Wert auf das Ergebnis hat. Buchstaben lassen sich auch zur Vereinfachung verwenden, wie z. B.: π, ohne die häufig verwendete Zahl 3,14159265 immer wieder eingeben zu müssen.

Wir unterscheiden drei Arten von Variablen: einfache Variable, Feld-Variable und r-Variable. Feld-Variable werden auf Seite 44 näher beschrieben.

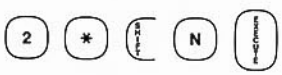
Einfache Variable

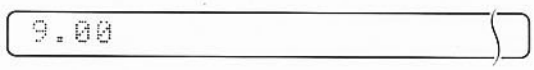
Bis 26 einfache Variable lassen sich mit dem Modell 9825A verwenden. Es sind dies die Großbuchstaben von A bis Z.

Um einer Variablen einen Wert zuzuordnen, wird die Taste → gedrückt. Durch Drücken der folgenden Tasten wird z. B. N der Wert 4,5 zugeordnet.

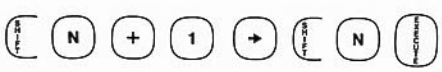
Die Zahl steht immer links vom Zuordnungspfeil und die Variable immer rechts.

Nun kann N in einer Berechnung eingesetzt werden. So wird z. B. N mit 2 multipliziert:





Der Wert der Variablen N ändert sich dadurch nicht. Der Variablen N können auch neue Werte zugeordnet werden, wie zum Beispiel:



r-Variable

r-Variable werden mit „r" und einer Zahl bezeichnet. Auch bei den Modellen 9820 und 9821A wurden r-Variable verwendet, zur Programmkompatibilität werden sie auch bei Modell 9825A verwendet.

Im folgenden Beispiel wird der Wert 12 r10 zugeordnet; dann wird der Wert 20 dem durch den Wert von r10 bestimmten Register zugeordnet (indirekte Speicherung).

```
12→ r10
20→rr10
```

der Wert 12 wird r10 direkt zugeordnet

der wert 20 wird r12 indirekt zugeordnet

Weitere Hinweise über r-Variable finden Sie auf Seite 45.

Betriebsarten

Der Rechner kann in drei Betriebsarten arbeiten: Tastenfeld-Modus, Programm-Modus, „Live Keyboard"-Modus.

- Bei Bedienung über das Tastenfeld werden keine Programme abgearbeitet.
- In Betriebsart „Program" werden Programme abgearbeitet.
- Bei der Betriebsart „Live Keyboard" können während der Ausführung von Programmen zwischenzeitlich Berechnungen über das Tastenfeld ausgeführt werden.

Korrektur-Tasten

Unterläuft Ihnen bei der Eingabe einer Zeile in die Anzeige ein Fehler, so kann dieser Fehler mit diesen Zeichen-Korrekturtasten berichtigt werden.

Wenn Sie z. B. diese Zeile eingeben wollen:

10 → A; 12 → B

aber folgendes eingeben:

```
10→a:112→B
```

Zur Korrektur brauchen Sie nur die Taste [BACK] so lange zu drücken, bis die blinkende Marke ▒ über dem „a" liegt.

Darauf werden die Umschalttaste und Ā gedrückt. Um die „1" in „112" zu löschen, wird einmal [FWD] und dann [DELETE] gedrückt. Die Anzeige sieht dann wie folgt aus:

```
10→A:12→B
```

wobei die blinkende Marke auf der „1" der 12 liegt. Um die Zeile auszuführen,

drücken: [ENTER]

In einem weiteren Beispiel wollen Sie folgendes berechnen:

$$10 + 18 + 22$$

aber Sie geben fälschlicherweise ein:

```
10+8+22
```

Um die 1 vor der 8 einzufügen, wird die Taste [BACK] viermal gedrückt. Das blinkende Austauschzeichen ▒ steht jetzt auf der 8. Dann wird die [INS/RPL] Taste gedrückt, dadurch wird das Austauschzeichen durch das Einfügungszeichen ◀ ersetzt. Jetzt wird die 1 eingegeben und die Anzeige sieht wie folgt aus:

```
10+18+22
```

Das Einfügungszeichen ◀ blinkt noch über der 8, wodurch angezeigt wird, daß weitere Zeichen eingefügt werden könnten. Um die Zeile auszuführen, wird [ENTER] gedrückt.

Programmierung

Zur Erstellung eines Programms wird in fünf Schritten vorgegangen:

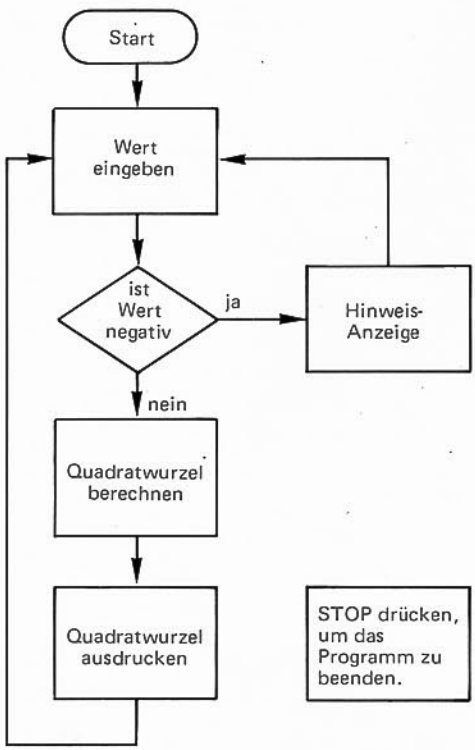
1. Das Problem definieren
2. Den besten Lösungsweg finden
3. Die Befehle für das Programm schreiben
4. Die Befehle in den Rechner eingeben
5. Das Programm korrigieren und abarbeiten.

Schritt 1:

Als einfaches Beispiel wollen wir die Quadratwurzel eines jeden eingegebenen Wertes ausdrucken lassen. Ist der eingegebene Wert negativ, sollte ein Hinweis ausgedruckt werden und das Programm weiterlaufen.

Schritt 2:

Es ist allgemein üblich, ein Flußdiagramm anzufertigen. Mit den am Ende dieses Abschnitts erklärten Symbolen zeichnen wir das Flußdiagramm:



Schritt 3:

Nach diesem Flußdiagramm werden die Befehle für das Programm geschrieben.

Programm	Beschreibung
„start": ent V	Der Variablen V wird der eingegebene Wert zugewiesen
if V < Ø; dsp „neg. V"; gto „start"	Entscheidung, ob V negativ, wenn ja, Sprung zum Anfang
√ V → S	S ist die Quadratwurzel
prt S	Ausdruck des Ergebnisses
gto „start"	Zurück zum Anfang, um neuen Wert einzugeben.

Beachten Sie, daß die zweite Zeile drei durch Semikolon getrennte Befehle enthält.

Schritt 4:

Jetzt wird [ERASE] [A] gedrückt, um den Speicher des Rechners zu löschen und dann das obige Programm Zeile für Zeile eingegeben. Nach jeder Zeile wird [STORE] gedrückt, um diese Zeile abzuspeichern.

Schritt 5:

Nach Speicherung des Programms wird [LIST] [SHIFT] gedrückt, um einen Programmausdruck zu erhalten. Um des Programm ablaufen zu lassen, wird [RUN] gedrückt. Wenn V ? erscheint, wird ein Wert eingegeben und [CONTINUE] gedrückt. Der Rechner druckt dann das Ergebnis aus.

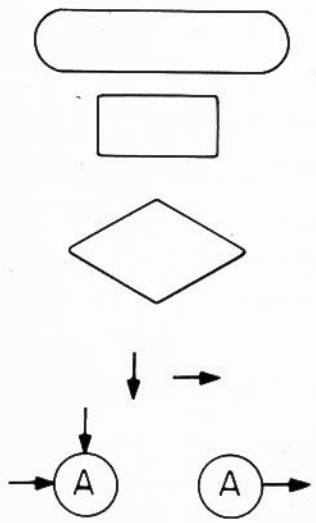
Bei positiven Werten läuft das Programm wie erwartet ab. Wird ein negativer Wert eingegeben, erfolgt eine nicht erkennbare, weil viel zu kurze Anzeige von neg. V, da das Programm sofort weiterläuft und wiederum V ? angezeigt wird.

Deshalb wird ein `wait`-Befehl nach dem Anzeigebefehl (dsp) in Zeile 1 eingefügt: Um das Programm zu ändern wird [FETCH] [1] [SHIFT] gedrückt und durch mehrmaliges Drücken der [BACK] Taste wird das blinkende Austauschzeichen vor das Semikolon des gto-Befehls gebracht. Jetzt wird [INS/RPL] gedrückt und `;wait 500` eingetastet. Mit [STORE] wird die Zeile 1 abgespeichert.
Ausdruck des abgeänderten Programms:

```
0: "start":ent V
1: if V<0;dsp
 "neg.V";wait
 500;gto "start"
2: rV→S
3: prt S
4: gto "start"
```

Bevor ein weiteres Programm eingegeben wird, wird `erase a` eingegeben und die Taste [SHIFT] gedrückt. Dadurch wird der Speicher des Rechners gelöscht.

In Flußdiagrammen übliche Symbole



Bedeutung

Ende oder Anfang des Programms

Teil eines Programms, meist ein Befehl

Dieses Entscheidungssymbol bedeutet eine Entscheidung für eine Verzweigung. Meist wird ein „if"-Befehl zur Entscheidung verwendet.

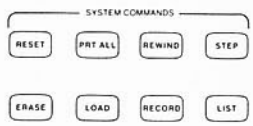
Richtungspfeile geben die Ablaufrichtung an.

Verbindungen, bedeutet Verbindung der zu- und abgehenden Linien.

ABSCHNITT 4
DAS TASTENFELD

Beim Lesen dieses Abschnitts sollte die Abbildung des Tastenfeldes am Ende des Handbuchs herausgeklappt werden. Zahlen, Anweisungen und Befehle können über das alphanumerische Tastenfeld eingegeben werden. Außerdem gibt es Tastengruppen zur Steuerung des Rechnersystems, für die Anzeige, Korrekturtasten für Zeichen und Zeilen, frei definierbare Funktionstasten und arithmetische Tasten. Außer der Schreibmaschinentastatur werden alle Tastengruppen in diesem Abschnitt erklärt.

Systemtasten



RESET Über diese Taste werden Rechner und die I/O-Karten in den Einschalt-Zustand versetzt, ohne daß Programme oder Variable gelöscht werden. Dieser Befehl wird beim Drücken sofort automatisch ausgeführt. Der Rechner hält dann sofort an, wenn ein Programm läuft und die Zeilennummer, an der das Programm gestoppt wird, wird angezeigt. Kann ein Programmablauf durch CLEAR oder STOP nicht unterbrochen werden, so wird RESET gedrückt.

PRT ALL Über diese Taste wird die Betriebsart „PRINT ALL" ein- und ausgeschaltet. Wird die Taste einmal gedrückt, erscheint on in der Anzeige. Alle gespeicherten Zeilen sowie eingegebene Formeln und angezeigte Ergebnisse werden dann ausgedruckt. Wird ein Programm abgearbeitet, werden alle angezeigten Hinweise und Fehlermeldungen ausgedruckt.

Durch nochmaliges Drücken dieser Taste wird „PRINT ALL" abgeschaltet, darauf erscheint Off in der Anzeige.

REWIND Nach Drücken dieser Taste wird die Bandkassette auf den Bandanfang zurückgespult. Um weitere Befehle auszuführen, braucht nicht gewartet zu werden, bis das Band zurückgespult ist.

STEP Mit dieser Taste läßt sich ein Programm Zeile für Zeile abarbeiten. Auf jeden Tastendruck wird die nächste Programmzeile ausgeführt, wobei die Nummer der nächsten Zeile angezeigt wird. Wird STEP das erste Mal in einem Programm gedrückt, so wird die Zeilennummer der auszuführenden Zeile angezeigt. Wird nochmals STEP gedrückt, wird die Zeile ausgeführt.

ERASE Mit dieser Taste können Teile oder der gesamte frei verfügbare Speicher (Read/Write) gelöscht werden.

ERASE A EXECUTE	Löscht den Gesamtspeicher
ERASE V EXECUTE	Löscht nur Variable
ERASE K EXECUTE	Löscht alle frei definierten Funktionen
ERASE EXECUTE	Löscht Programme und Variable
ERASE fn	Löscht die frei definierte Funktionstast„n"

Die Tabelle in Anhang D enthält alle Funktionen, die durch die „erase a" und „erase"-Anweisungen beeinflußt werden.

LOAD Mit dieser Taste werden Programme oder Daten von der Kassette in den Rechner geladen, Beispiel:

LOAD 3 EXECUTE	Dabei wird das Programm von file 3 in den Rechner geladen.

In der Anzeige erscheint ldf „load file", wenn diese Taste gedrückt wird. Eine ausführliche Beschreibung finden Sie auf Seite 121.

RECORD Mit dieser Taste werden Programme und Daten auf der Bandkassette gespeichert. Bevor auf der Bandkassette gespeichert werden kann, müssen „files" markiert werden (siehe „mark"-Befehl [mrk] auf Seite 108).

Im folgenden Beispiel wird angenommen, daß Files schon markiert sind.

RECORD 6 EXECUTE	Das Programm wird in File 6 auf der Kassette gespeichert.

In der Anzeige erscheint `rcf` „record file" wenn diese Taste gedrückt wird.

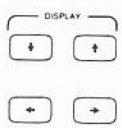
Eine ausführliche Beschreibung zur Speicherung von Programmen und Daten finden Sie unter dem „record file"-Befehl auf Seite 115.

LIST Durch Drücken dieser Taste können vollständige Programme, Programmteile, die Funktionen aller oder einzelner frei definierbarer Tasten ausgedruckt werden.

Beispiel:

LIST FUNCTION	Druckt das gesamte Programm aus.
LIST K FUNCTION	Druckt die Funktionen aller frei definierbaren Tasten in numerischer Reihenfolge aus.
LIST f_0	Druckt die Funktionen der Taste f_0 aus.
LIST 2 0 FUNCTION	Druckt das Programm von Zeile 20 bis Programmende aus.
LIST 9 , 1 3 FUNCTION	Druckt das Programm von Zeile 9 bis 13 aus.

Tasten zur Steuerung der Anzeige



↑ Durch Druck auf diese Taste wird die vorhergehende Zeile eines Programms angezeigt. Wird ein Stop-Befehl im Programm ausgeführt oder wird eine Zeilennummer angezeigt, so wird diese Zeile durch Drücken der Taste ↑ angezeigt.

Nach einer Fehlermeldung wird die Zeile, die den Fehler enthält, mit der Taste ↑ angezeigt.

↓ Mit dieser Taste wird die nächstfolgende Zeile eines Programms angezeigt. Folgt keine weitere Zeile im Programm, wird die Anzeige gelöscht und weitere Zeilen können dem Programm hinzugefügt werden.

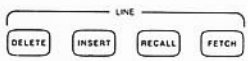
[←] Mit dieser Taste kann eine Zeile in der Anzeige nach links geschoben werden, um die gesamte Zeile darzustellen. Jedesmal, wenn die Taste gedrückt wird, springt die Zeile um 8 Zeichen nach links. Steht das Markierungszeichen in der Anzeige, bleibt es stehen, wenn die [←] Taste gedrückt wird.

[→] Mit dieser Taste wird die Anzeige nach rechts verschoben, sie entspricht sonst der vorher beschriebenen Taste.

Korrekturtasten

Es gibt zwei Arten Korrekturtasten; zur Korrektur von Zeilen und zur Korrektur von Zeichen.

Tasten zur Korrektur von Zeilen



[FETCH] Mit dieser Taste können Zeilen und die Funktionen der frei definierbaren Tasten angezeigt werden.

Beispiel:

[FETCH] [2] [0] [ENDLINE] Die Zeile 20 wird angezeigt.

[FETCH] [f4] Die Funktion der Taste f_4 wird angezeigt.

Ist Taste 4 nicht definiert, wird f4 angezeigt.

[DELETE] Mit dieser Taste kann die angezeigte Programmzeile gelöscht werden. Wird keine Zeile angezeigt, ertönt ein Signal und die Taste bleibt unwirksam. Um eine Zeile zu löschen, wird die Zeile in die Anzeige geholt und [DELETE] gedrückt. Wird eine Zeile innerhalb eines Programms gelöscht, so werden die Adressen aller relativen und absoluten go to und go sub Befehle automatisch umnumeriert.

Diese Taste unterscheidet sich von der später beschriebenen Löschtaste für einzelne Zeichen. Zur Löschung mehrerer Programmzeilen kann die „delete" (del) Anweisung verwendet werden, die auf Seite 92 beschrieben ist.

INSERT Mit dieser Taste kann eine neue Zeile vor einer in die Anzeige geholten Zeile eingefügt werden. Eine Zeile kann über folgende Tasten in die Anzeige gebracht werden: FETCH ↑ ↓

Beispiel: Einfügen der Zeile ⌜A→B zwischen Zeile 20 und 21.

Drücken: FETCH 2 1

Eingabe: ⌜A→B

Drücken: INSERT

```
20: A+1→A
21: sto 25
```

```
20: A+1→A
21: ⌜A→B
22: sto 26
```

Wird eine Zeile in ein Programm eingefügt, so werden die Verzweigungsadressen aller relativen und absoluten go to und go sub Befehle entsprechend umnumeriert.

RECALL Mit dieser Taste können die letzte oder die vorletzte Eingabe über das Tastenfeld angezeigt werden. Durch einmaliges Drücken von RECALL wird die letzte Eingabe, durch zweimaliges Drücken die vorletzte Eingabe angezeigt.

Diese Taste wird auch häufig verwendet, wenn Sie arithmetische Ausdrücke über das Tastenfeld errechnen und das Ergebnis nochmals mit dem Ausdruck verglichen werden soll. Bei vielen Arten von Fehlern wird der Fehler selbst durch eine blinkende Marke gekennzeichnet.

Tasten zur Korrektur von Zeichen

CHARACTER: BACK FWD DELETE INS/RPL

Zeilen, die mit den Tasten ↑ ↓ RECALL oder über die FETCH-Anweisung in die Anzeige geholt wurden oder eingetippte Zeilen, können durch Korrektur von einzelnen Zeichen berichtigt werden.

Zwei blinkende Zeichen lassen sich mit dieser Taste steuern: Das Austauschzeichen ▒ und das Einfügungszeichen ⌟ .

BACK Mit dieser Taste wird das blinkende Einfügungs- oder Austauschzeichen in der Anzeige nach links geschoben. Wird kein blinkendes Zeichen angezeigt, so wird durch Drücken von BACK das Zeichen auf das Ende der Zeile gesetzt.

FWD Mit dieser Taste wird das angezeigte Zeichen nach rechts geschoben. Ist noch kein blinkendes Zeichen sichtbar, so erscheint nach Druck der FWD Taste das Zeichen am Anfang der Zeile.

DELETE Mit dieser Taste können einzelne Zeichen, die mit einem blinkenden Zeichen gekennzeichnet sind, gelöscht werden. Diese Taste darf nicht mit der Taste zur Löschung von Zeilen verwechselt werden.

INS RPL Mit dieser Taste können die beiden blinkenden Zeichen ausgetauscht werden. Die Tasten BACK und FWD dienen zum Verschieben der blinkenden Zeichen. Wird das Einfügungszeichen angezeigt, erscheinen über die Tastatur eingetippte Zeichen links vom blinkenden Einfügungszeichen. Wird das Austauschzeichen angezeigt, erscheint das eingegebene Zeichen ebenfalls links vom blinkenden Zeichen.

Die unter dem Austauschzeichen liegenden Zeichen werden durch die Neueingaben ersetzt.

Tasten zur Steuerung des Rechners

RUN Durch Druck auf diese Taste wird ein im Rechner gespeichertes Programm, beginnend ab Zeile Ø, abgearbeitet. Der Befehl dieser Taste wird sofort ausgeführt. Alle Variable, flags und Unterprogramm-Rücksprungadressen werden dabei gelöscht. Wird ein Programm abgearbeitet, so wird dies durch das rote Lämpchen links in der Anzeige angezeigt.

Die Tabelle in Anhang D enthält alle Funktionen, die durch die Taste RUN beeinflußt werden.

STORE Mit dieser Taste werden Programmzeilen gespeichert. Bei der Zuordnung von Funktionen für die frei definierbaren Tasten werden auch diese Funktionen mit STORE gespeichert. Eine Programmzeile kann aus einem oder mehreren durch Semikolon getrennte Befehle bestehen. Tritt bei der Speicherung einer Zeile ein Fehler auf, so wird diese Zeile durch RECALL wieder angezeigt.

Meist wird dabei der Fehler durch ein blinkendes Zeichen markiert.

SHIFT und SHIFT LOCK Mit diesen Umschalttasten wird die Schreibmaschinentastatur auf Großbuchstaben und Sonderzeichen umgeschaltet. Wird die Taste SHIFT LOCK gedrückt, leuchtet ein Lämpchen über der Taste auf. Um die Taste „shift lock" zu löschen, wird SHIFT gedrückt.

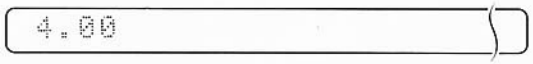
STOP Mit dieser Taste wird ein Programm am Ende der gerade abgearbeiteten Zeile angehalten. Dabei wird die Nummer der nächsten Programmzeile angezeigt. Wird (STOP) gedrückt, werden alle Enter-, List-, t-list- und Wait-Befehle ignoriert, die vollständige Zeile aber noch ausgeführt. Wird (STOP) in einem Eingabebefehl gedrückt, wird flag 13 gesetzt und der Eingabebefehl ist beendet.

Außerdem gibt es einen Stop-Programmbefehl, der auf den Seiten 57 und 89 beschrieben ist.

EXECUTE Mit dieser Taste wird die angezeigte Zeile mit einem oder mehreren Befehlen ausgeführt. Die beiden letzten über die Tastatur ausgeführten Befehle oder Ausdrücke sind vorübergehend gespeichert und können durch einmaliges oder zweimaliges Drücken von (RECALL) wieder angezeigt werden.

Das Ergebnis einer numerischen Berechnung über die Tastatur, das nicht einer Variablen zugeordnet ist, wird im Ergebnisspeicher gespeichert (siehe (RESULT) Taste).

Beispiel: (2) (+) (2) (EXECUTE)



Durch Drücken der Execute-Taste wird das Ergebnis angezeigt und im Ergebnisspeicher (Result) gespeichert.

Obwohl Ausdrücke wie: 2+2:r4 (EXECUTE)

berechnet werden können, wird jedoch nur das Ergebnis der letzten Berechnung der Zeile angezeigt und in den Ergebnisspeicher übernommen. Ist der Drucker auf „print all" geschaltet, werden beide Ergebnisse ausgedruckt.

CONTINUE Durch Drücken dieser Taste läuft ein Programm an der Stelle weiter, an der es gestoppt wurde. Wird eine Zeilennummer angezeigt, so läuft das Programm mit dieser Zeilennummer weiter, falls nicht (RESET) gedrückt wurde oder Programm-Korrekturen vorgenommen wurden. Dann startet das Programm mit Zeile Ø, wenn (CONTINUE) gedrückt wird.

Nach einem Eingabebefehl wird [CONTINUE] gedrückt, nachdem die Werte eingegeben sind. Werden keine Werte eingegeben und [CONTINUE] gedrückt, so behält die Variable ihren vorherigen Wert und flag 13 wird gesetzt. Wird der Programmablauf durch einen Fehler gestoppt und danach „CONTINUE" gedrückt, so bedeutet das, daß das Programm wieder beginnend mit Zeile Ø abgearbeitet wird.

Siehe auch „Continue"-Anweisung auf Seite 92.

[RESULT] Mit dieser Taste wird das Resultat einer numerischen Berechnung über die Tastatur von dem Ergebnisregister in die Anzeige zurückgerufen, falls das Ergebnis nicht einer Variablen zugewiesen wurde.

Beispiel:

Drücken: [3] [*] [6]

```
3*6
```

Drücken: [R/S]

```
18.00
```

Das Ergebnis 18 wird angezeigt und im Ergebnisspeicher abgespeichert. Mit dem Inhalt dieses Ergebnisspeichers kann dann weitergearbeitet werden, wie z.B.:

Drücken: [RESULT] [/] [2]

```
res/2
```

Drücken: [R/S]

```
9.00
```

In einem Programm können Werte nicht dem Ergebnisspeicher zugeordnet werden, jedoch können Werte vom Ergebnisspeicher Variablen zugeordnet oder in Berechnungen verwendet werden.

Beispiel:

```
0: 20→res
1: res+2→A
```

Dies ist nicht zulässig

Dadurch wird der Wert im Ergebnisspeicher + 2 der Variablen A zugeordnet.

CLEAR Mit dieser Taste wird die Anzeige gelöscht. Wird diese Taste bei der Abfrage einer Eingabe gedrückt, erscheint ein Fragezeichen, womit angezeigt wird, daß eine Eingabe erforderlich ist. Wird diese Taste nach der Anzeige der Funktion einer frei definierbaren Taste gedrückt, so wird deren Bezeichnung (z. B. f8) angezeigt.

→ Mit dieser Zuordnungstaste werden Werte Variablen zugeordnet. Dies ist nicht die gleiche Taste, die zur Steuerung der Anzeige verwendet wird.

Beispiel:

Drücken: SHIFT ↑ 5 → SHIFT X STORE

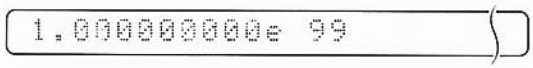
Dadurch wird die Quadratwurzel von 5 in X gespeichert.

I π Diese Taste wird gedrückt, um den Wert π einzugeben. Der eingegebene Wert ist 3,14159265360.

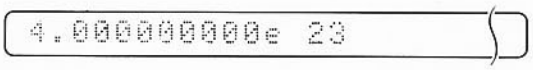
(−) ENTER EXP Nach Druck auf diese Taste erscheint ein e in der Anzeige und gibt damit einen Exponenten zur Basis 10 an. Auch die Taste E (ohne Umschalttaste) kann anstelle der Taste (−) ENTER EXP verwendet werden.

Beispiel:

Drücken: 1 (−) ENTER EXP 9 9 STORE



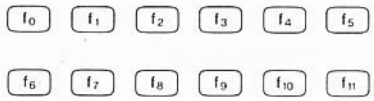
Drücken: 4 E 2 3 STORE



Anzeige:

Es ist zu beachten, daß zwischen den Funktionen der Taste (−) ENTER EXP und E hier kein Unterschied besteht.

Frei definierbare Funktionstasten



Diese speziellen Tasten, f0 bis f11 können als Programmierhilfen, als direkt ausführbare Funktionstasten oder zur sofortigen Weiterführung von Programmen verwendet werden. Diesen 12 Tasten lassen sich mit der Umschalttaste insgesamt 24 spezielle Funktionen zuordnen.

Um einer Taste eine bestimmte Funktion zuzuordnen, wird zuerst die Taste FETCH und dann die gewünschte Funktionstaste gedrückt. Darauf wird die zuzuordnende Zeile eingegeben. Mit der Taste STORE wird diese Funktion unter der entsprechenden Taste abgespeichert.

Beispiel:

Drücken: FETCH f0	Ist der Taste keine Funktion zugeordnet, wird `f0` angezeigt.
Eintippen: `prt`	`prt` wird angezeigt.
Drücken: STORE	Für f_0 wird `prt` als Eingabehilfe gespeichert.

Soll keine Funktion für die Taste gespeichert werden, so kann die Eingabe auch mit STOP beendet werden.

Um den Inhalt einer frei definierbaren Funktionstaste auszudrucken, wird LIST und dann die entsprechende Funktionstaste gedrückt.

Soll der Inhalt aller frei definierbaren Funktionstasten in numerischer Reihenfolge ausgedruckt werden, wird `list k` eingetippt und EINGABE gedrückt.

Unmittelbare Ausführung der speziellen Funktionen

Steht vor der unter einer Taste abgespeicherten Zeile ein Sternchen, wird diese Funktion nach Drücken der Taste sofort abgearbeitet.

Beispiel:

Drücken: [FETCH] [SHIFT] [f11]

Ruft f_{23} auf (f_{11} mit Umschalttaste)

Eintippen: `*prt "π";π`

Durch das Sternchen wird die Zeile sofort nach Drücken der Taste ausgeführt.

Drücken: [STORE]

Dadurch wird die eingegebene Zeile unter f_{23} gespeichert.

Wird jetzt [SHIFT] [f11] gedrückt und die Anzeige war frei, so wird „π" mit dem entsprechenden Wert ausgedruckt.

```
π              3.14
```

Direkt auszuführende Funktionen unter einer Funktionstaste werden oft verwendet, um bestimmte Programmteile innerhalb eines Programms aufzurufen und abzuarbeiten. Mit dem Continue-Befehl und einer Zeilennummer können verschiedene Verzweigungen direkt ausgeführt werden. Als Beispiel werden [f0] und [f10],wie folgt belegt:

[f0] `*cont 5`

[f10] `*cont 10`

Wird jetzt [f0] gedrückt, läuft das Programm mit Zeile 5, wird [f10] gedrückt, mit Zeile 10 weiter.

Sofortige Weiterführung von Programmen mit den speziellen Funktionstasten

Steht vor dem einer speziellen Taste zugeordneten Wert ein Schrägstrich (/), so wird bei einem Enter-Befehl durch Drücken der Funktionstaste der zugeordnete Wert eingegeben und das Programm weitergeführt.

Das bedeutet, daß die „Continue"-Anweisung automatisch erfolgt, wenn die Funktionstaste gedrückt wird. Dies wird zur Eingabe oft benutzter Werte verwendet.

Beispiel:

Drücken:	FETCH f_{10}	Ruft Taste f_{10} auf.
Eintippen:	/2.71828182846	Dadurch wird der Wert „e" eingegeben.
Drücken:	STORE	Damit wird die angezeigte Zeile unter f_{10} gespeichert.

Wird jetzt ein Wert von einem Enter-Befehl abgefragt und wird die Taste f_{10} gedrückt, so wird „e" eingegeben (2.71828282846) und das Programm läuft weiter.

Zuordnung von mehreren Befehlen unter eine Taste

Werden Einzelbefehle durch Semikolon getrennt, so lassen sich auch diese Befehlsfolgen unter einer Taste abspeichern. Ein Beispiel ist die Umwandlung von Zoll in Zentimeter.

Die folgende Zeile wird für die Taste f_0 gespeichert.

Drücken: [FETCH] [f_0]

Eintippen: `*→R;dspR;"in.= 2.54R; cm."`

Drücken: [STORE]

Jetzt geben wir 6 ein und drücken [f_0] . Folgendes erscheint in der Anzeige:

```
6.00in.=  15.24 cm
```

ABSCHNITT 5
PROGRAMMBEFEHLE, FUNKTIONEN UND OPERATOREN

Alle in diesem Abschnitt behandelten Befehle, Funktionen und Operatoren sind programmierbar, fast alle können auch bei Bedienung über das Tastenfeld ausgeführt werden.

Befehle können programmiert oder ausgeführt werden. Operatoren und Funktionen müssen Teil eines Befehls sein, um programmiert zu werden. Das bedeutet, daß Operationen, wie z. B. 10 + 32 oder $\sqrt{63}$, die über das Tastenfeld berechnet werden, Teil eines Befehls sein müssen, um sie zu programmieren. Deshalb sind `10+32→X` oder `prt√63` zulässige Befehle.

Syntax

Allen in diesem Handbuch behandelten Befehlen liegt eine feste Syntax zugrunde. Eine Liste finden Sie im Anhang A dieses Handbuchs.

[ ] — Angaben in Klammern können verwendet werden, falls gewünscht.

Gepunktete Angaben — Gepunktete Angaben müssen wie angegeben erscheinen.

. . . — 3 Punkte zeigen an, daß die vorhergehende Angabe wiederholt werden kann.

Variable

Es können drei verschiedene Arten von Variablen verwendet werden: einfache Variable, Feld-Variable und r-Variable. Alle Variablen haben den Anfangswert 0. Elemente von einfachen Variablen, Feld-Variablen oder r-Variablen belegen jeweils 8 Bytes im Speicher.

Einfache Variable

Es können 26 einfache Variable von A bis Z verwendet werden. Einfache Variable werden durch Großbuchstaben gekennzeichnet. Jeder Variablen läßt sich ein Wert zuordnen.

Beispiel:

```
A: 12→A
1: prt A
```

Ordnet A den Wert 12 zu.
Druckt den Wert von A aus.

Feld-Variable

Es lassen sich 26 Felder von A bis Z verwenden.

Der Bezeichnung der Felder folgen die Elemente in eckigen Klammern (z. B. L[31]).

Ein Feld muß in einem Dimensionierungs-Befehl angegeben sein. Dadurch wird Speicherplatz für das Feld reserviert und allen Elementen 0 zugeordnet.

Im Dimensionierungs-Befehl müssen die Dimensionen des Feldes angegeben werden. Wird nur die obere Grenze angegeben, wird als untere Grenze 1 angenommen, ist die untere Grenze nicht 1, müssen beide Werte angegeben werden.

Beispiel:

```
dim A[4,5]
```

Reserviert Speicherplatz für die 20 Elemente des zweidimensionalen Feldes A.

```
dim P[-2:1,-2:2]
```

Reserviert Speicherplatz für die 20 Elemente des zweidimensionalen Feldes P. (Obere und untere Grenze sind angegeben.)

Ein Feld kann beliebig groß sein und eine beliebige Anzahl Elemente enthalten, wenn die Zeilenlänge und die Speicherkapazität nicht überschritten wird. Werden die Grenzen festgelegt, so müssen sie innerhalb −32767 und +32767 liegen.

Ein einzelnes Element eines Feldes wird mit seiner Position spezifiziert.

Beispiel:

```
4→A[1,5]
```

4 wird dem Element 1,5 des Feldes A zugewiesen.

```
3→P[-2,1]
```

3 wird dem Element –2,1 des Feldes P zugewiesen.

Noch ein Beispiel:

```
0: dim Q[10,10]
1: 3→Q[7,1]
```

Reserviert 100 Speicherplätze für Feld Q. Q(7,1) wird der Wert 3 zugeordnet.

```
2: 5→Q
```

Der Wert 5 wird der einfachen Variablen Q zugeordnet. Zwischen der einfachen Variablen Q und dem Feld Q [10,10] besteht kein Zusammenhang.

```
3: 2→Q[1,Ø]
```

Q (1,5) wird der Wert 2 zugeordnet.

r-Variable

r-Variable werden durch ein kleines r, dem ein Wert oder ein Ausdruck folgt, angegeben. Wird eine r-Variable verwendet, so wird Speicherplatz für alle niedrigeren r-Variablen reserviert, die nicht zugeordnet sind. Werden r-Variablen verwendet, erhalten sie den Wert Ø. Wird r10 ein Wert zugeordnet, so werden r0 bis r9 automatisch reserviert. Sie erhalten den Wert Ø, wenn keine anderen Werte zugeordnet wurden. Jede r-Variable belegt 8 Byte des Speichers.

Beispiel:

```
0: 4→r0
1: 2→rr0
```

4 wird der r-Variablen Ø zugeordnet.
2 wird der r-Variablen 4 zugeordnet,
r0 = 4, deshalb wird 2→r4 zugeordnet. Dies wird als indirekte Speicherung verstanden.

Speicherplatz für Variable

Einfachen und r-Variablen wird automatisch Speicherplatz zugewiesen, wenn ein Befehl mit der jeweiligen Variablen ausgeführt wird. Feld-Variablen wird Speicherplatz mit einem Dimensionierungsbefehl reserviert.

Vor der Zuordnung von Speicherplatz für Variable werden folgende 3 Bedingungen geprüft:

1. Bevor eine Variable mit einem Dimensionierungsbefehl zugeordnet wird, wird geprüft, ob eine Zuordnung schon erfolgt ist. Wenn ja, erfolgt eine Fehlermeldung und das Programm hält an.

2. Wenn einfache Variable in mehreren Befehlen enthalten sind, wird geprüft, ob sie bereits in anderen Befehlen zugeordnet wurden. Falls sie noch nicht zugeordnet sind, erfolgt eine Zuordnung.

3. Wenn ein Feldelement in mehreren Befehlen enthalten ist, wird geprüft, ob das Feld bereits dimensioniert wurde, falls noch nicht dimensioniert, erfolgt eine Fehlermeldung.

Innerhalb eines Befehls werden Variable von links nach rechts festgelegt, wie sie auftreten.

Zahlenformate

Zahlen lassen sich mit Fest- oder Gleitkomma darstellen. Intern arbeitet der Rechner immer mit der vollen Stellenzahl, so daß die Genauigkeit immer erhalten bleibt.

Wird der Rechner eingeschaltet oder (RESET) gedrückt oder `erase a` ausgeführt, schaltet sich der Rechner automatisch auf „fixed 2". Das bedeutet 2 Stellen nach dem Dezimalpunkt.

Fixed-Befehl

Syntax:

`fxd` [Zahl der Stellen nach dem Komma]

Mit dem fixed (fxd)-Befehl wird das Format für die Anzeige oder den Ausdruck festgelegt. In der Fixed-Angabe wird die Stellenzahl nach dem Komma festgelegt. Es lassen sich 0 bis 11 Stellen vorgeben.

Ist eine Zahl für dieses Format zu groß, so wird kurzzeitig auf das vorher verwendete Gleitkommaformat umgeschaltet. Deshalb gilt für die Zahl A:

$A = N^* 10^E$

dabei ist $1 \leqslant N < 10$, oder $N = 0$

Die Anzeige schaltet auf Gleitkomma um, wenn

D + E > 14

dabei sind:

D = Anzahl der Nachkommastellen, im fixed-Befehl angegeben

E = Exponent.

Im folgenden Beispiel wird die Umschaltung auf die vorhergehende „float 9"-Einstellung gezeigt. Folgendes Programm eingeben:

```
0: ent A
1: fxd 0;prt A
2: fxd 1;prt A
3: fxd 2;prt A
4: fxd 3;prt A
5: end
```

Wird der Wert `125e10` eingegeben, wenn `A?` erscheint, wird folgendes ausgedruckt:

```
  1250000000000
1250000000000.0
1.250000000e 12
1.250000000e 12
```

Sind Zahlen zu klein, um in das Format zu passen, erscheinen in allen Stellen Nullen, wobei bei negativen Zahlen das Minuszeichen erscheint.

Beispiel:

```
0: fxd 3;dsp -
 .000125;wait
 2000
1: fxd 2;dsp
 .000204
```

```
-0.000
```

```
 0.00
```

Hier einige Zahlen und ihr Format, wenn fixed 3 vorgegeben ist:

Zahlen	Darstellung
18	18.000
–.000006	-0.000
–2.7532	-2.753
4.5678	4.568
5.3111 e3	5311.100
1234567891234.5	1.234567891e 12
	(vorher float 9)

Float-Befehl

Syntax:

`flt` [Anzahl der Nachkommastellen]

Mit dem float (flt)-Befehl wird Gleitkomma-Anzeige vorgegeben. Dies ist besonders bei sehr großen und sehr kleinen Zahlen von Vorteil. Float 0 bis float 11 kann angegeben werden.

Die Anzeige bei Gleitkomma sieht wie folgt aus:

—D.D...De—DD

- Führende Nullen werden unterdrückt.

  Bei negativen Zahlen erscheint das Minuszeichen vor der ersten Ziffer. Bei positiven Zahlen oder 0 bleibt die Stelle frei.

- Nach der ersten Ziffer erscheint der Dezimalpunkt, außer bei flt 0.

- Hinter dem Punkt können weitere Stellen folgen, wobei ihre Anzahl durch den Formatbefehl festgelegt ist (flt 5 = 5 Stellen hinter dem Komma).

- Dann folgt das Zeichen „e", darauf das Minuszeichen oder eine Leerstelle (bei positiven Exponenten) und die zwei Stellen des Exponenten zur Basis 10. Durch den Exponenten mit Vorzeichen wird angegeben, um wieviel Stellen der Dezimalpunkt nach links oder rechts verschoben werden muß.

Hier einige Zahlen, wenn `flt 2` eingestellt ist:

Zahlen	Darstellung
—3.2	`-3.20e 00`
271	`2.71e 02`
26.377	`2.64e 01`
.000004	`4.00e-6`
2.482e33	`2.48e 33`

Anzahl der Stellen

Unabhängig vom eingestellten Format werden alle Zahlen intern mit 12 Stellen gespeichert. Als Beispiel wird `fxd5` und dann folgende Zahl eingegeben:

```
123456789.56789
```

Nach Drücken der [ENTER] -Taste wird folgendes angezeigt:

```
123456789.56700
```

Die 13. und 14. Stelle ist nicht gespeichert und an ihrer Stelle werden Nullen angezeigt.

Runden

Ist die Zahl länger als die Anzahl der eingestellten Nachkommastellen, wird die letzte angezeigte Ziffer gerundet. Ist die erste nicht mehr angezeigte Stelle gleich oder größer 5, wird die letzte angezeigte Stelle aufgerundet, ist diese Zahl kleiner 5, wird abgerundet. Die interne Genauigkeit bleibt jedoch bestehen.

Beispiel:

```
0: fxd 2
1: dsp 1.235;
 wait 1000
2: dsp 2.404
3: end
```

```
1.24
```

```
2.40
```

Display-Befehl

Syntax:

`dsp` [beliebige Kombination von Text und Ausdrücken]

Mit dem Display (dsp)-Befehl können Werte oder Text angezeigt werden. Variable und Text werden durch Kommas getrennt (z. B. `dsp"No.",N,B`).

Alpha-Text wird mit Anführungszeichen gekennzeichnet. Sollen Anführungszeichen auch dargestellt werden, müssen [und [2] zweimal gedrückt werden.

Beispiel:

Eintippen: `dsp"Say""Hi""to her."`

Drücken: [ENTER]

Anzeige:

```
Say"Hi"to her.
```

Die Zahl der angezeigten Zeichen wird durch die Stellen der Anzeige bestimmt. Ist eine Zeile länger als 32 Zeichen, so kann der Rest der Zeile mit den Tasten [←] und [→] angezeigt werden.

Werte und Text bleiben so lange in der Anzeige stehen, bis sie durch eine weitere Operation (wie Enter (ent)) gelöscht werden oder ein print (prt)-Befehl ausgeführt wird.

Print-Befehl

Syntax:

`prt` [beliebige Kombination von Ausdrücken oder Text]

Mit dem print (prt)-Befehl werden Werte oder Text über den eingebauten Drucker ausgedruckt.

Beispiel:

Befehl	Ausdruck
`prt6`	`6.00`
`prt"One";1`	`One 1.00`
`prt"This one"`	`This one`

Soll z. B. folgender Ausdruck gedruckt werden: `prt 6*7→X`

so wird der Befehl ausgeführt und der entsprechende Wert ausgedruckt (und wie hier, in X gespeichert).

Sollen Anführungszeichen in einem Text gedruckt werden, wird für jedes Anführungszeichen (SHIFT) (") zweimal gedrückt.

Beispiel:

Eintippen: `prt"Ent""1"" or ""0"""`

Drücken: (ENTER)

Anzeige:
```
Ent "1" or "0"
```

Variable oder Text werden durch Kommas getrennt.

Beispiel:

Eintippen: `prt"First",1,"Next",2`

Drücken: (ENTER)

Ausdruck:
```
First          1.00
Next           2.00
```

Der Ausdruck von Text und Zahlen erfolgt nach folgenden Regeln:

- Texte mit einer folgenden Zahl werden auf einer Zeile ausgedruckt, wenn der Platz reicht. Andernfalls wird der Text in einer Zeile und die Zahl auf der nächsten Zeile ausgedruckt.

- Ist Text durch Kommas getrennt, wird jedesmal eine neue Zeile angefangen, wobei auf der nächsten Zeile weitergeschrieben wird, wenn der Text mehr als 16 Zeichen hat.

- Werden Zahlen durch Kommas getrennt, so wird jede Zahl auf einer neuen Zeile geschrieben, wenn nicht flt 10 oder flt 11 eingestellt ist, wobei 2 Zeilen zum Ausdruck erforderlich sind.

Die Eingabe `prt` ohne nachfolgende Parameter, hat auf den Drucker keinen Einfluß.

Enter-Befehl

Syntax:

`ent` [Bezeichnung ,] Variable [, [Bezeichnung ,] Variable ...]

Mit dem Enter-Befehl (ent) werden bei laufenden Programmen Variablen über das Tastenfeld Werte zugeordnet. Diese Variable können einfache Variable, Feld-Variable oder r-Variable sein.

Beispiel:

```
4: ent""Q
5: ent A,B[3],
 r11
```

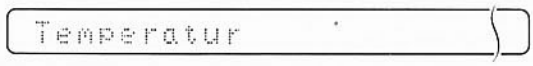
Stoppt das Programm bei einem Enter-Befehl, wird der Wert eingegeben und (CONTINUE) gedrückt.

Werden viele Werte über das Tastenfeld eingegeben, so ist es eine große Hilfe, die Bezeichnung der Variablen, der ein Wert zugeordnet werden soll, in der Anzeige darzustellen.

Beispiel

```
0: ent "Betrag",
 B
1: ent "Temperat
 ur",T
```

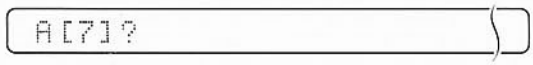
Anzeige



Wird kein Text zur Bezeichnung verwendet, setzt der Rechner die Variable als Bezeichnung ein.

Beispiel:

```
3: ent A[7]
```



Wird ein nur mit Anführungszeichen bezeichnetes Feld als Bezeichnung eingegeben, wie `10: ent "",A` , so bleibt die vorherige Anzeige stehen, wenn nicht der Anzeigebefehl und der Eingabebefehl durch einen Druckbefehl getrennt werden. Dies ist von Vorteil bei variablen Bezeichnungen in einem Enter-Befehl.

Beispiel:

```
7: 1976→Y:fxd 0
8: dsp "Juli,",Y
9: ent "",A
```

Während das Programm auf eine Eingabe wartet, lassen sich Berechnungen über das Tastenfeld ausführen. Es brauchen nur die Werte eingegeben und die Execute-Taste gedrückt zu werden. Sollen die Ergebnisse dieser Berechnungen in das Programm eingegeben werden, werden bei Zahlen [RECALL] oder [RESULT] und dann [CONTINUE] gedrückt. Wird [ENTER] gedrückt, so ist dies gleichbedeutend als ob [CONTINUE] ohne Eingabe von Werten gedrückt würde.

Auf eine Abfrage können auch komplexe Zeilen eingegeben werden.

Beispiel:

```
0: ent B
1: ent A
2: prt A
3: end
```

Bei der Abfrage:

```
B?
```

Wird ein Wert für B eingegeben.

Bei der Abfrage von:

```
A?
```

eintippen: `20;ifP>20;40`

Darauf [CONTINUE] drücken. Ist der eingegebene Wert für B größer als 20, wird `40` ausgedruckt, ist er kleiner, wird `20` gedruckt.

Wird [CONTINUE] gedrückt, ohne einen Wert einzugeben, behält die Variable ihren ursprünglichen Wert und flag 13 wird gesetzt. Wird ein Wert eingegeben, wird flag 13 gelöscht (siehe flags, Seite 69).

Um ein Programm bei einer Eingabe abzubrechen, wird [STOP] gedrückt. Der Rest der Zeile wird jedoch noch ausgeführt.

Bei der Eingabe (Enter) dürfen keine Anweisungen, wie `fetch` oder `run` , eingegeben werden (error 03).

Das folgende Beispiel zeigt eine Besonderheit bei der Eingabe des `ent`-Befehls. Arbeiten Sie dieses kurze Programm ab.

```
0: dim A[20]
1: 4→I
2: ent I,A[I]
```

```
I?
```

Eintippen: 8

```
A[4]?
```

Drücken: (CONTINUE)

Hierbei wird der frühere Wert von „I" nach dem Enter-Befehl verwendet und nicht der eingegebene Wert von „I". Um den eingegebenen Wert von „I" als Element zu verwenden, wird die Zeile 2 im obigen Programm wie folgt geändert:

```
2: ent I;ent
 A[I]
```

Ein Enter-Befehl kann mehrere Variable beinhalten. Es kann jedoch jeweils nur ein Wert eingegeben werden.

Beispiel:

```
0: ent A,B
```

Auf A? wird der Wert für A eingegeben, dann (CONTINUE) gedrückt, worauf B? in der Anzeige erscheint und eingegeben wird.

Enter-Print-Befehl

Syntax:

enp [Bezeichnung ,] Variable [, [Bezeichnung ,] Variable ...]

Der Enter-Print-Befehl (enp) entspricht dem Enter-Befehl, wobei jedoch hier die Bezeichnung und der eingegebene Wert angezeigt und ausgedruckt werden.

Als Beispiel hier ein kurzes Programm zur Berechnung des Inhalts eines Kreises:

```
0: enp "Radius",
 R
1: πRR→F
2: prt "Flaeche"
 ,F
3: end
```

Wird im Programm 2 für „R" eingegeben, wird folgendes ausgedruckt:

```
Radius
2
Flaeche     12.57
```

Space-Befehl

Syntax:

`spc` [Anzahl der Leerzeilen]

Auf den Space-Befehl (spc) transportiert der Drucker um die angegebene Zahl der Leerzeilen weiter. Die Zeilenzahl kann im Bereich von 0 bis 32767 liegen. Erfolgt keine Angabe, wird eine Leerzeile eingefügt.

Beispiel:

```
0: spc A+B

1: spc 5
2: spc
```

Die Anzahl der Leerzeilen wird durch A + B bestimmt.
5 Leerzeilen
1 Leerzeile

Befehl für akustischen Hinweis

Syntax: `beep`

Auf den Beep-Befehl gibt der Rechner ein akustisches Signal.

Bei der Quadratwurzel einer negativen Zahl gibt der Rechner normalerweise ein akustisches Signal und zeigt `error 67` an. Im folgenden kurzen Programm wird der eingegebene Wert für A geprüft. Ist der Wert negativ, ertönt ein Piepton und ein Hinweis erscheint, aber vom Programm werden weitere Werte abgefragt.

```
0: fxd 4
1: "start":ent
 "Argument";A
2: if A<0;gto
 "error"
3: prt √A;gto
 "start"
4: "error":beep
5: dsp "√ of
 neg. no."
6: wait 2000
7: gto "start"
```

Wait-Befehl

Syntax:

`wait` Anzahl der Millisekunden

Auf den Wait-Befehl hält das Programm für die angegebene Anzahl der Millisekunden an. Dieser Wait-Befehl wird oft zur Anzeige von Werten, zur Abfrage von Eingaben oder zur Anzeige von Hinweisen verwendet. Die Anzahl der Millisekunden ist anzugeben.

Da auch der Wait-Befehl eine gewisse Zeit zur Ausführung benötigt, erfolgt die Anzeige bei kleinen Werten länger als angegeben. Die größte Zeitspanne beträgt 33 s, eingegeben durch die Zahl 32767.

Beispiel:

```
wait2000
2: wait 2*I
```

Dauer von 2 s
Dauer von (2*I) ms

Im folgenden Beispiel folgt ein Eingabebefehl auf den Anzeigebefehl. Die Anzeige wird mit dem Wait-Befehl eine Sekunde stehengelassen:

```
9: dsp "Bitte
 eingeben";wait
 1000
10: ent "Betrag
 von X",X
```

„Bitte eingeben" bleibt etwa eine Sekunde in der Anzeige stehen, dann erst erscheint die nächste Anzeige.

Stop-Befehl

Syntax: `stp`

Mit dem Stop-Befehl wird das Programm am Ende der Zeile, in der die Anweisung vorkommt, gestoppt. Dies gilt auch für über das Tastenfeld eingegebene Zeilen. Bei einem Stop-Befehl in einem Programm wird die Zeilennummer der nächsten Zeile angezeigt. Durch Drücken der Taste (CONTINUE) läuft das Programm ab dieser Zeilennummer weiter. Mit (STEP) kann das Programm Zeile für Zeile durchgetastet werden. Werden nach einem Programmstop Programmänderungen vorgenommen, so läuft nach Drücken der Taste (CONTINUE) oder (STEP) das Programm wieder mit Zeile 0 an.

Der Stop-Befehl kann auch zur Programmkorrektur verwendet werden. Weitere Hinweise finden Sie im Abschnitt über Programmkorrektur.

End-Befehl

Syntax: `end`

Der End-Befehl ist meist die letzte Zeile eines Programms. Das Programm hält dann wie beim Stop-Befehl an. Durch den Endbefehl wird der Programmzähler jedoch auf Zeile 0 gesetzt. Außerdem werden alle Unterprogramm-Rücksprungadressen gelöscht (siehe: go sub-Befehl).

Der End-Befehl kann nicht auf einen Enter-Befehl oder bei der Betriebsart „Live keyboard" ausgeführt werden.

Hierarchie

Enthält ein Befehl Funktionen, arithmetische Operationen, Vergleichsbefehle, logische Operationen, eingefügte Zuordnungen oder flags, so werden diese Operationen in einer bestimmten Reihenfolge nach folgender Hierarchie ausgeführt:

höchste Priorität	Funktionen, flags, r-Variable
↓	↑ (Potenzieren)
↓	Implizierte Multiplikation
↓	– Vorzeichen (minus)
↓	*/mod
↓	+–
↓	Logische Vergleiche (=,>,<,<=>=,#)
↓	not
↓	and
geringste Priorität	or, xor

Ein Ausdruck wird von links nach rechts abgearbeitet. Jeder Operator wird mit dem rechts stehenden Operator verglichen. Hat der Operator rechts höhere Priorität, wird er wiederum mit dem von ihm rechts stehenden verglichen. Dies wird so lange wiederholt, bis ein Operator gleicher oder niedrigerer Priorität ermittelt wird. Dann wird die Operation mit der höchsten Priorität oder die erste Operation von den zwei gleichwertigen ausgeführt. Darauf wiederholt sich dieser Vorgang des Abarbeitens von links nach rechts. Werden Klammern festgestellt, so werden die Ausdrücke in den Klammern ebenfalls nach ihrer Priorität abgearbeitet. Im folgenden Beispiel sind S_1, S_2 usw. Zwischenergebnisse.

2A=B+C–DmodEexp(notF)	Implizierte Multiplikation
S_1 = B+C–DmodEexp(notF)	Addition
S_1 = S_2 –DmodEexp(notF)	Klammern auflösen
S_1 = S_2 –DmodEexp S_3	Exponentialfunktionen
S_1 = S_2 –DmodE S_4	Implizierte Multiplikation
S_1 = S_2 –Dmod S_5	Modulus
S_1 = S_2 – S_6	Subtraktion
S_1 = S_7	Gleichsetzung
S_8	Endergebnis

Operatoren

Die vier Gruppen mathematischer oder logischer Symbole, Operatoren genannt, sind: der Zuordnungs-Operator, die arithmetischen Operatoren, Vergleichs-Operatoren und logische Operatoren.

Zuordnungs-Operator

Syntax:

Ausdruck → Variable

Mit dem Zuordnungs-Operator → werden Variablen Werte zugeordnet.

Beispiel:

```
1.4→A
3: B→A
```

Der Wert 1,4 wird der Variablen A zugeordnet.
Der Wert B wird der Variablen A zugeordnet.

Außerdem können noch Werte mit dem Enter-Befehl (ent) oder dem load file-Befehl (ldf) Variablen zugeordnet werden.

Um mehreren Variablen denselben Wert zuzuordnen, wird der Zuordnungs-Operator wie in diesem Beispiel verwendet:

```
32→A→B→X→r4
```

Dies kann auch in der folgenden Form erfolgen:

`(25→A)+1→B` dies entspricht: `25→A:A+1→B`
Dabei ist zu beachten, daß `25→A+1→B` nicht zulässig ist. Eingefügte Zuordnungen müssen in Klammern stehen.

Zuordnungen können auch wie folgt in einen Befehl eingefügt werden:

`if(A+1→A)>5` Dadurch kann die Zuordnung und der Vergleich mit einem Befehl erfolgen.

Arithmetische Operatoren

Die folgenden sechs arithmetischen Operatoren können verwendet werden:

+	Addition	A+B or +A
−	Subtraktion	A−B or −A
*	Multiplikation	A*B
/	Division	A:B
↑	Potenzieren	A^B
mod	Modulus	A mod B ist der Rest von A:B, wenn A und B Integerzahlen sind A−int (A/B)*B.

Neben dem * für die Multiplikation kann die Multiplikation impliziert sein. In den folgenden Beispielen wird automatisch multipliziert:

- Wenn 2 Variable hintereinander stehen (wie AB).
- Wenn ein Wert und eine Variable hintereinander stehen (wie 5A).
- Wenn eine Variable oder ein Wert vor der Klammer steht (wie 5(A+B)).
- Wenn Klammern hintereinander stehen [wie: (A+B) (X+Y)].
- Wenn eine Variable, ein Wert oder eine Klammer vor einer Funktion stehen (wie 32 sin A).

Beispiel:

AB→X	A mal B wird in X gespeichert.
(5)5→X	5 mal 5 wird in X gespeichert.
A(B+C)→B	A mal Summe B+C wird in B gespeichert.
5absB	5 mal Absolutwert von B.

Die Reihenfolge, in der arithmetische Operationen ausgeführt werden, wird unter „Hierarchie" erklärt.

Vergleichs-Operatoren

Die folgenden sechs Vergleichs-Operatoren können verwendet werden:

Symbole	Bedeutung	
=	gleich	
>	größer als	
<	kleiner als	
= > oder > =	größer oder gleich	Die verschiedenen Darstellungsformen sind gleichwertig.
= < oder < =	kleiner oder gleich	
# oder <> oder ><	ungleich	

Das Ergebnis einer Vergleichsoperation ist entweder 1 (wenn der Vergleich wahr ist) oder 0 (wenn der Vergleich nicht wahr ist). Ist A kleiner als B, ist der Ausdruck A<=B wahr und das Ergebnis ist 1. Alle Vergleiche werden mit 12 Stellen, Vorzeichen und Exponenten durchgeführt.

Bei Zwischenergebnissen darf der Exponent zwischen –511 und +511 liegen.

Vergleichs-Operatoren können in Zuordnungs-Befehlen, if-Befehlen und allen Befehlen, in denen Ausdrücke als Argumente zulässig sind, verwendet werden.

Beispiele:

```
A=B→C
```
Zuordnungs-Befehl, wenn A und B gleich sind, wird 1 in C gespeichert, andernfalls wird 0 in C gespeichert.

```
if A>B; ...
```
if-Befehl. Ist A größer als B, wird die Zeile weiter abgearbeitet. Ist A kleiner oder gleich B, wird zur nächsten Zeile gesprungen.

```
... ;jmpA>3
```
Sprung-Befehl. Ist A größer als 3, wird eine Zeile übersprungen. Andernfalls Sprung zum Anfang der Zeile zurück (jmp 0).

```
prtA(A>B)+B(A<B)
```
Druck-Befehl. Ist A größer als B, wird A gedruckt. Ist A kleiner als B, wird B gedruckt. Ist A gleich B, wird 0 gedruckt.

Die Reihenfolge, in der Vergleichs-Operatoren abgearbeitet werden, wird unter „Hierarchie" erklärt.

Logische Operatoren

Die vier logischen Funktionen AND, OR, XOR (exklusives oder) und NOT werden in der Boolschen Algebra verwendet. Jeder Wert außer Null (falsch) wird als 1 (wahr) angesehen. Das Ergebnis einer logischen Verknüpfung ist entweder 0 oder 1. Die Reihenfolge, in der die logischen Operationen ablaufen, wird unter „Hierarchie" erklärt.

Operation	Syntax	Wahrheitstabelle		
AND	Ausdruck `and` Ausdruck	A	B	A und B
		F	F	0
		F	W	0
		W	F	0
		W	W	1
OR	Ausdruck `or` Ausdruck	A	B	A oder B
		F	F	0
		F	W	1
		W	F	1
		W	W	1
XOR (exklusives ODER)	Ausdruck `xor` Ausdruck	A	B	A xor B
		F	F	0
		F	W	1
		W	F	1
		W	W	0
NOT	`not` Ausdruck	A		not A
		F		1
		W		0

Beispiel:

Programm:

```
0: .1→A;0→B
1: prt "A and
 B",A and B
2: prt "A or B",
 A or B
3: prt "A xor
 B",A xor B
4: prt "not A",
 not A
5: end
```

Ausdruck:

```
A and B      0.00
A or B       1.00
A xor B      1.00
not A        0.00
```

Mathematische Funktionen und Befehle

Mathematische Funktionen und Befehle werden in dem folgenden Abschnitt erklärt.

In den Beispielen finden Sie Klammern nur, wenn sie wirklich erforderlich sind.

Allgemeine Funktionen

Syntax	Beschreibung	Beispiele (fxd 5)
√ Ausdruck	Berechnet die Quadratwurzel eines positiven Ausdrucks. Bei negativen Ausdrücken siehe Fehlerhinweise auf Seite 67	√64 = 8.00000 √π = 1.77245
abs	Berechnet den Absolutwert eines Ausdrucks.	abs(−3.09) = 3.09000 abs 330.1 =330.10000
sgn	Diese Funktion ergibt −1 bei negativen Ausdrücken, 0 wenn der Ausdruck 0 ist und 1 bei positiven Ausdrücken.	sgn(−18) = −1.00000 sgn34 = 1.00000 sgn0 = 0.00000
int	Integer-Funktion ergibt die größte Ganzzahl, die kleiner oder gleich dem Ausdruck ist.	int2.718 = 2.00000 int(−3.24) = −4.00000
frc	Ergibt den gebrochenen Anteil einer Zahl. (Bei negativen Zahlen wird der Integerwert von der ursprünglichen Zahl abgezogen).	frc2.718 = .71800 frc(−3.24) = .76000
prnd	Rundet auf die nächste Zehnerpotenz; das Argument bleibt unverändert.	prnd(127.375,−2) = 127.38000 127.375 wird auf der (10^{-2}) Stelle gerundet.
drnd	Rundet das Argument auf die angegebene Stellenzahl. Das Argument bleibt unverändert.	drnd(73.0625,5) = 73.06300 drnd(−65023,1) = −70000.00000 drnd(.055,1) = 0.06000
min	Ruft den kleinsten enthaltenen Wert auf. Das Minimum in einem Feld kann durch ein Sternchen wie B[*] ermittelt werden.	0:dimA[3]:2→A[1] 1:9→A[2]:3→A[3] min(A[*]) = 2.00000 min(2,−3,−3,4) = −3.00000

Syntax	Beschreibung	Beispiele (fxd 5)
max	Ruft den größten enthaltenen Wert auf. Das Maximum in einem Feld kann durch ein Sternchen wie z. B. [*] ermittelt werden.	0:dimA[3];2→A[1] 1:9→A[2];3→A[3] max(A[*])= 9.00000 max(5,4,-3,8) = 8.00000
rnd	Die Funktion für Zufallszahlen liefert Pseudo-Random-Zahlen im Bereich ⩽0 < bis 1. Ist das Argument positiv wird die Anfangszahl als π/180 (0,0174532925200) festgelegt. Diese Zahl wird aufgerufen, wenn der Rechner eingeschaltet wird, „erase a" ausgeführt wird, oder (RESET) gedrückt wird. Jeder weitere Zugriff mit einem positiven Argument baut auf dem vorhergehenden Argument auf. Um eine andere Anfangszahl als π/180 zu erhalten, benützt man ein negatives Argument. Der gebrochene Anteil des Absolutwertes des Arguments wird als Anfangszahl verwendet. Um geeignete Anfangszahlen zu finden, wird ein Ausdruck zwischen 0 und −1 genommen: Je weniger Nullen enthalten sind, um so besser, die letzte Stelle sollte 1, 3, 7 oder 9 sein.	rnd1 = 0.67822 0:wait rnd (-.31317) 1:rnd1→A Beachten Sie, daß der wait-Befehl anstatt des Zuordnungsbefehls verwendet wird, um die Anfangszahl festzulegen. Zeile 1 generiert eine Zufallszahl, die auf die Anfangszahl 0.31317 aufgebaut ist und nicht auf π/180.

Exponential- und logarithmische Funktionen

Syntax	Beschreibung	Beispiele (fxd 5)
ln	Der natürliche Logarithmus eines positiven Ausdrucks.	ln8001 = 8.98732 ln.0026 = –5.95224
exp	Die Exponentialfunktion potenziert die Konstante „e" mit dem Ausdruck. Der Bereich des Arguments liegt zwischen –227,95 und 230,25.	exp1 = 2.71828 exp(–3) =.04979
log	Berechnung des Log (Basis 10) eines positiven Ausdrucks.	log305.2 = 2.48458 log.0049 = 2.30980
tn↑	Dabei wird die Konstante 10 mit dem Ausdruck potenziert. Das Argument kann im Bereich –99 und 99.999 liegen. Diese Funktion wird schneller als 10↑ ausgeführt.	5tn↑2 = 500.00000 tn↑(–3) = .00100

Mathematische Fehler und automatisch zugewiesene Werte bei log- und ln-Funktionen werden auf Seite 67 beschrieben.

Trigonometrische Funktionen und Befehle

Die sechs trigonometrischen Funktionen werden im jeweils eingestellten Winkelmaß berechnet, die mit den drei in diesem Abschnitt beschriebenen Befehlen eingestellt werden.

Altgrad wird beim Einschalten des Rechners, wenn „erase a" ausgeführt oder RESET gedrückt wird, eingestellt.

deg — Alle Funktionen werden in Altgrad berechnet (Kreisbogen gleich 360 Grad).

rad — Alle Berechnungen werden in Bogenmaß ausgeführt (Kreisbogen sind 2π Bogenmaß).

grad — Alle Berechnungen werden in Neugrad ausgeführt (Kreisbogen ist 400 Neugrad).

units — Zeigt das eingestellte Winkelmaß an.

Syntax	Beschreibung	Beispiele (fxd 5)
`sin`	Berechnet den Sinus im eingestellten Winkelmaß.	`deg:sin45` = 0.70711 `rad:sin(π/6)` = 0.50000 `grad:sin(-70)` =0.89101
`cos`	Berechnet den Cosinus im eingestellten Winkelmaß.	`deg:cos45` = 0.70711 `rad:cos(π/6)` = 0.86603 `grad:cos(-70)` = 0.45399
`tan`	Berechnet den Tangens im eingestellten Winkelmaß.	`deg:tan45` = 1.00000 `rad:tan(π/4)` = 1.00000 `grad:tan50` = 1.00000
`asn`	Berechnet den Arcus-Sinus im eingestellten Winkelmaß. Der Bereich des Arguments ist −1 bis +1. Das Ergebnis liegt bei Bogenmaß zwischen $-\pi/2$ und $+\pi/2$, bei Altgrad zwischen −90 und +90, bei Neugrad zwischen −100 und +100.	`deg:asn.8` = 53.13010 `rad:asn.8` = 0.92730 `grad:asn.8` = 59.03345
`acs`	Berechnet den Arcus-Cosinus im eingestellten Winkelmaß. Bereich des Arguments −1 bis +1. Bei Bogenmaß liegt das Ergebnis zwischen 0 und π, 0 und 180 bei Altgrad −100 und +100 bei Neugrad.	`deg:acs(-.4)` = 113.57818 `rad:acs(-.4)` = 198231 `grad:acs(-.4)` = 126.19798
`atn`	Berechnet den Arcus-Tangens im eingestellten Winkelmaß. Das Ergebnis liegt bei Bogenmaß zwischen $-\pi/2$ und $+\pi/2$, −90 und +90 bei Altgrad, −100 und +100 bei Neugrad.	`deg:atn20` = 87.13759 `rad:atn20` = 1.52084 `grad:atn20` =96.81955

Mathematische Fehler

Bei mathematischen Fehlern erscheint „error 66" bis „error 77" in der Anzeige. Diese Fehler werden in diesem Abschnitt beschrieben.

Tritt ein mathematischer Fehler auf, wird flag 15 automatisch gesetzt. Ob ein Fehler angezeigt wird, hängt davon ab, ob flag 14 gesetzt ist. Ist flag 14 gesetzt, so ergeben mathematische Operationen, bei denen normalerweise eine Fehlermeldung angezeigt würde, bestimmte Werte.

Sollen diese Werte gedruckt, angezeigt oder gespeichert werden, so werden sie automatisch in ±9.99999999999e 99 geändert.

error 66 Division durch Null. Der Wert ist 9.99999999999e 511, wenn der Divident positiv ist, –9.99999999999 511, wenn der Divident negativ ist.
Beispiel: `-9.5/0` = –9.99999999999e 511
A mod B, wobei B gleich Null ist, der Wert ist Null wie in diesem Beispiel:

```
32mod0 = 0
```

error 67 Quadratwurzel einer negativen Zahl. Der zugewiesene Wert ist $\sqrt{\text{abs (Argument)}}$.

Beispiel: `r(-36)` = 6.

error 68 tan von (nxπ/2 Bogenmaß);
tan von (nx90 Altgrad);
tan von (nx100 Neugrad);
wobei „n" eine ungerade Ganzzahl ist. Der Wert ist
9.99999999999e 511, wenn „n" positiv ist und
–9.99999999999e 511, wenn „n" negativ ist.

Beispiel:

```
rad:tan(-π/2)     = –9.99999999999e 511
deg:tan270        = 9.99999999999e 511
grad:tan300       = 9.99999999999e 511.
```

error 69 In oder log einer negativen Zahl. Der zugewiesene Wert ist ln von (abs (Argument)) oder log (abs (Argument)).

Beispiel:

```
ln(-301)    = 5.70711
log(-.001) = -3.00000
```

error 79 In oder log von Null. Der Wert ist –9.99999999999e 511.

Beispiel:

```
ln0   = -9.99999999999e 511
log0 = -9.99999999999e 511
```

error 71 arc sin oder arc cos einer Zahl < –1 oder > 1.
Der zugewiesene Wert ist asn (sgn (Argument)) oder acs (sgn (Argument)).

Beispiel (in Grad):

```
deg:asn(-10) = -90
deg:acs(1.5) =   0
```

error 72 Negative Basis mit gebrochenem Exponenten. Der zugewiesene Wert ist (abs (Basis)) ↑ (nicht ganzzahlige Potenz).

Beispiel:

```
(-36)↑(.5)  = 6
```

error 73 Null, mit Null potenziert. Der zugewiesene Wert ist 1.

error 74 Zahl ist für Speicher zu groß. Der zugewiesene Wert ist 9.99999999999e 99 oder –9.99999999999e 99.

Beispiel:

```
(1e62)*(1e38)→A   A gleich 9.99999999999e 99.
(-1e25)↑5→B       B gleich –9.99999999999e 99.
```

error 75 Zahl ist zu klein. Der zugewiesene Wert ist 0.

Beispiel:

```
(1e-66)*(4e-35)→A
```

A gleich 0

error 76 Zwischenergebnis ist zu groß. Der zugewiesene Wert ist 9.99999999999e 511 oder –9.99999999999e 511.

Beispiel:

```
(1e99)↑6          = 9.99999999999e 511
(-1e99)↑7         = –9.99999999999e 511.
```

error 77 Zwischenergebnis ist zu klein. Der zugewiesene Wert ist 0.

Beispiel:

```
(1e10)↑60         = 0
```

Flags

Flags sind programmierbare Hinweise, die den Wert 1 oder 0 haben können. Ist eine Flag gesetzt, ist ihr Wert 1, ist die Flag gelöscht, so ist ihr Wert 0. Der Wert einer Flag kann mit dem Complement-Flag-Befehl (cmf) umgekehrt werden. Insgesamt gibt es 16 flags von 0 bis 15. Die folgenden Flags haben spezielle Bedeutung:

Flag 13 – wird automatisch gesetzt, wenn nach einem Enter-Befehl (STOP) oder (CONTINUE) gedrückt wird, ohne Werte einzugeben. Flag 13 wird gelöscht, wenn Werte eingegeben werden.

Flag 14 – ist Flag 14 gesetzt, so ignoriert der Rechner mathematische Fehler wie Division durch Null und zeigt den in der Liste auf Seite 67 zugewiesenen Wert an.

Flag 15 – wird automatisch gesetzt, wenn ein mathematischer Fehler erscheint, unabhängig von Flag 14.

Set Flag-Befehl

Syntax:

sfg [Flag-Nummer , ...]

Der set flag-Befehl (sfg) setzt den Wert der in der Liste angegebenen Flags auf 1. Die Flag-Nummer kann ein Wert oder ein Ausdruck sein. Wird sfg allein ausgeführt, werden alle Flags (0 bis 15) gesetzt. Wird eine nicht ganzzahlige Flag angegeben, so wird ihr Wert zu einer Ganzzahl gerundet.

Beispiel:

```
sfg2
0:sfgA+1
1:sfg1,X
```

Flag 2 setzen.
Die durch (A+1) bestimmte Flag setzen.
Flag 1 und die durch X bezeichnete Flag setzen.

Clear Flag-Befehl

Syntax:

cfg [Flag-Nummer , ...]

Der Clear Flag-Befehl (cfg) setzt die angegebenen Flags auf 0. Die Flag-Nummer kann ein Wert oder ein Ausdruck sein.

Wird cfg allein ausgeführt, werden alle Flags (0 bis 15) auf 0 gesetzt. Wird eine nicht ganzzahlige Flag angegeben, so wird ihr Wert zu einer Ganzzahl gerundet.

Beispiel:

```
cfg14
3:cfg flg2

4:cfg
```

Flag 14 löschen.
Die durch den Wert von Flag 2 bestimmte Flag löschen (entweder Flag 1 oder Flag 0 werden gelöscht).
Alle Flags löschen.

Complement Flag-Befehl

Syntax:

`cmf` [Flag-Nummer [, ...]]

Durch den Complement Flag (cmf)-Befehl wird der Wert der angegebenen Flags geändert. Dadurch wird eine gesetzte Flag zu 0. Eine gelöschte Flag wird durch den Complement Flag-Befehl zu 1. Als Flag-Nummer kann ein Wert oder ein Ausdruck verwendet werden. Zur Umkehr der Flags 0 bis 15 wird der Complement Flag-Befehl ohne Parameter verwendet. Wird eine nicht ganzzahlige Flag angegeben, so wird ihr Wert zu einer Ganzzahl gerundet.

Beispiel:

```
    cmf 1
 0:cmf X-1
 1:cmf 3,4,5
```

Flag 1 umkehren.
Die durch (X—1) bezeichnete Flag umkehren.
Flags 3, 4 und 5 umkehren.

Flag-Funktion

Syntax:

`flg` [Flag-Nummer]

Mit der Flag (flg-Funktion) wird der Wert einer Flag geprüft. Das Ergebnis ist 1 oder 0. 1 zeigt eine gesetzte Flag, 0 eine gelöschte Flag an.

Beispiel:

```
 4:if flg2:jmp5
 5:flg15→A
```

Ist Flag 2 gesetzt, 5 Zeilen überspringen.
Ist Flag 15 gesetzt, dann 1→A.
Ist Flag 15 gelöscht, dann 0→A.

Verzweigungsbefehle

Mit Verzweigungsbefehlen wird der kontinuierliche Programmablauf verändert. Dadurch wird es möglich, einen Teil des Programms in mehreren Schleifen zu durchlaufen, Unterprogramme auszuführen und aufgrund von logischen Entscheidungen zu verschiedenen Programmteilen zu verzweigen. Zur Verzweigung dienen drei Befehle: der go to (gto)-Befehl, der jump (jmp)-Befehl und der go sub (gsb)-Befehl.

Die folgenden drei Arten der Verzweigung können bei go to und go sub verwendet werden:

Absolute Verzweigung –	automatische Verzweigung zur angegebenen Zeilennummer (wie `gto10`).
Relative Verzweigung –	dabei wird um die angegebene Zeilenzahl im Programm vor- oder zurückverzweigt (wie `gsb-3`).
Verzweigung zu einer Marke –	Verzweigung zur angegebenen Marke. Diese Verzweigung wird am häufigsten benutzt, da die Zeilennummer nicht bekannt sein muß (wie `gto "First"`).

Umnumerierung der Zeilen

Wenn eine Programmzeile eingefügt oder gelöscht wird, werden die übrigen Zeilen automatisch umnumeriert. Dabei werden auch Zeilennummern relativer oder absoluter go to- oder go sub-Befehle automatisch mitverändert. Bevor eine Löschung ausgeführt wird, prüft der Rechner das gesamte Programm. Ist die zu löschende Zeile die Adresse eines relativen oder absoluten go to- oder go sub-Befehls, wird ein Fehler angezeigt und die Zeile wird nicht gelöscht, bis ein Sternchen * in den Löschbefehl eingefügt wird (siehe Löschbefehle Seite 92).

Wird eine Zeile durch die Umnumerierung zu lang, so erscheint ihre Zeilennummer mit einem Fragezeichen, wenn die Zeile angezeigt oder ausgedruckt wird.

Beispiel:

```
8: ... ;gto99
```

Zeile 8 enthält 73 Zeichen.

Wird z. B. als Zeile 7 eine neue Zeile eingefügt, so wird Zeile 8 in 9 umnumeriert und die Sprungadresse von 99 auf 100 geändert. Es wird dann angezeigt:

```
 9: ?...
```

Um die gesamte Zeile anzeigen zu können wird eine niedrigere Zeilennummer gelöscht, um den alten Zustand wieder herzustellen. Auch wenn eine Zeile zu lang wird, so wird sie doch im Programm abgearbeitet.

Weitere Erklärungen über Umnumerierung finden Sie im Abschnitt 5.

Marken

Marken sind Zeichen oder Bezeichnungen in Anführungszeichen am Anfang der Zeile, nach einem gto- oder gsb-Befehl oder nach einem Run- oder Continue-Befehl. Hinter Marken am Zeilenanfang muß ein Doppelpunkt stehen.

Marken werden für Verzweigungen verwendet. Die hinter einem gto- oder gsb-Befehl angegebene Marke wird im Programm gesucht. Am Ende der Zeile wird zu der Zeile mit dieser Marke verzweigt. Soll erstmalig in einem Programm zu einer Marke verzweigt werden, wird das gesamte Programm, beginnend mit Zeile 0, nach der Marke abgesucht. Später wird dann direkt nach dieser Marke verzweigt. Bei der Suche nach Marken werden alle Zeichen, auch Leerstellen, verglichen.

Beim Schreiben von Programmen ist es oft erforderlich, Hinweise einzufügen. Wie im folgenden Beispiel können dafür Marken verwendet werden:

```
 7: "Area":
 8: πR↑2→A
```

Hinter einer Marke muß ein Doppelpunkt stehen, auch wenn darauf keine weiteren Befehle folgen.

Go to-Befehl

Mit dem go to (gto)-Befehl wird eine Verzweigung angegeben und das Programm an der bezeichneten Stelle weitergeführt. Enthält eine Zeile mehr als einen go to-Befehl, so wird nur der letzte Befehl ausgeführt.

Absolute go to-Befehle

Syntax:

`gto` Zeilennummer

Mit dem absoluten go to-Befehl wird zur angegebenen Zeilennummer verzweigt. Die Zeilennummer muß dabei eine ganze Zahl sein (wie 5 oder 13).

Wird ein absoluter go to-Befehl über das Tastenfeld ausgeführt, so wird der Programmzähler auf die angegebene Zeile gesetzt. Um diese Zeile anzuzeigen, wird [↑] gedrückt.

Relative go to-Befehle

Syntax:

`gto+` Anzahl der Zeilen
`gto-` Anzahl der Zeilen

Mit dem relativen go to-Befehl kann innerhalb eines Programms von der jeweiligen Zeile aus um die angegebene Zeilenzahl im Programm vor- (+) oder zurückgesprungen (—) werden. Die Zeilennummer muß ganzzahlig sein.

Beispiel:

```
20:gto +1
21:gto -3
22:gto +0
23:gto -0
```

1 Zeile weiter.
3 Zeilen zurück.

Zurück zum Zeilenanfang.

Go to zu einer Marke

Syntax:

```
gto        Marke
```

Bei einem go to-Befehl mit einer Marke wird zur angegebenen Marke verzweigt. Dies ist die einfachste Art der Verzweigung, da keine Zeilennummer bekannt sein muß.

Beispiel:

```
gto "Avg."
```
Verzweigung zur Zeile mit der Marke „Avg".

Wird ein go to-Befehl zu einer Marke über das Tastenfeld ausgeführt, so wird der Programmzähler auf die angegebene Zeile gesetzt. Um diese Zeile anzuzeigen, wird [↑] gedrückt.

In Verbindung mit if-Befehlen können mehrere gto-Befehle in einer Zeile verwendet werden, um nach logischen Entscheidungen nach verschiedenen Richtungen zu verzweigen. Dies wird auf Seite 80 erklärt.

Sprungbefehl

Syntax:

```
jmp        Anzahl der Zeilen
```

Mit dem jump (jmp)-Befehl kann um die angegebene Zeilenzahl vor- und zurückgesprungen werden. Dieser Befehl ist dem relativen go to-Befehl ähnlich, außer daß bei jmp die Anzahl der Zeilen ein Ausdruck sein kann. Ist die Anzahl der Zeilen positiv, wird im Programm weitergesprungen, ist die Anzahl der Zeilen 0, wird auf den Anfang derselben Zeile zurückgesprungen.

Ist die Anzahl negativ, wird im Programm zurückgesprungen. Ist die Zeilennummer keine Ganzzahl, wird sie bei 0,5 und größer aufgerundet, bei kleiner 0,5 abgerundet.

Der jump-Befehl wird langsamer als der go to-Befehl ausgeführt.

Der jump-Befehl kann nur am Zeilenende stehen, andernfalls wird Fehler 07 angezeigt, wenn die Zeile gespeichert werden soll.

Beispiele:

```
0:jmp 10
```

Sprung um 10 Zeilen vorwärts.

```
1:jmp A
```

Sprung um die Anzahl der Zeilen, die durch den Wert von A angegeben ist.

```
2:jmp(Z=2)2
```

Wenn Z gleich 2, zwei Zeilen weiterspringen, andernfalls zum Zeilenanfang zurück.

```
3:    ...:jmp(B+
1→B)>20
```

B inkrementiert, und wenn B größer als 20, zur nächsten Zeile springen, andernfalls zum Zeilenanfang zurückspringen.

Go to subroutine und return-Befehle

Mit dem go to subroutine (gsb)-Befehl kann in Unterprogramme verzweigt werden. Unterprogramme werden verwendet, wenn derselbe Programmteil mehrfach abgearbeitet und von verschiedenen Stellen im Programm aufgerufen werden soll. Wird ein go sub-Befehl ausgeführt, wird ein Rücksprungzeichen auf die nächste, dem gsb-Befehl folgende Zeile gesetzt. Mit dem return-Befehl (ret) im Unterprogramm läuft das Programm dann mit dieser Zeile weiter. Der return-Befehl muß der letzte in einem Unterprogramm ausgeführte Befehl sein und muß am Ende der Zeile stehen. Die Tiefe der go sub-Verschachtelung wird lediglich durch den Speicherplatz begrenzt.

Jedes Rücksprungzeichen belegt 8 Byte im Speicher. In Unterprogramme sollte nur über den go sub-Befehl verzweigt werden. Ein Rücksprung sollte nur über den ret-Befehl vorgenommen werden. Enthält eine Zeile mehr als einen go to- oder go sub-Befehl, so wird nur der letzte Befehl ausgeführt.

Es gibt drei Arten von go sub-Befehlen: absolute, relative und zu Marken.

Absolute go to-Befehle

Syntax:

`gsb` Zeilennummer

Mit dem absoluten go sub-Befehl wird zur angegebenen Zeilennummer des Unterprogramms verzweigt. Die Zeilennummer muß ganzzahlig sein.

Beispiel:

```
7:gsb 15
18:ret
```

Sprung in das Unterprogramm in Zeile 15.
Return-Befehl am Ende des Unterprogramms.

Relative go sub-Befehle

Syntax:

`gsb+` Anzahl der Zeilen
`gsb-` Anzahl der Zeilen

Mit einem relativen go sub-Befehl kann im Programm um die angegebene Zeilenzahl vor- (+) und zurückgesprungen (—) werden. Die Anzahl der Zeilen muß als ganze Zahl angegeben werden.

Beispiel:

```
7: gsb +5
8: gsb -3
```

Auf das Unterprogramm in Zeile 12 weiterspringen.
Auf das Unterprogramm in Zeile 5 zurückspringen.

Go sub zu Marken

Syntax:

```
gsb      Marke
```

Bei einem go sub-Befehl zu einer Marke wird auf das mit der Marke bezeichnete Unterprogramm verzweigt. Es ist dies die einfachste Form der Verzweigung mit dem go sub-Befehl, da keine Zeilennummer berücksichtigt werden muß.

Beispiel:

```
3: gsb "sub1"
```

Sprung in das Unterprogramm der mit der Marke „sub 1" bezeichneten Zeile.

In Verbindung mit dem if-Befehl können mehrere go sub-Befehle in einer Zeile verwendet werden, um nach logischen Entscheidungen in verschiedene Richtungen zu verzweigen. Dies wird auf Seite 80 erklärt.

Berechnete go sub-Verzweigung

Mit dem jump-Befehl, zusammen mit dem go sub-Befehl, sind berechnete Verzweigungen zu Unterprogrammen möglich.

Diese Befehle haben die folgende Form:

```
gsb  beliebige Adresse ;jmp  Ausdruck.
```

Die Zeilennummer oder die Marke in dem go sub-Befehl wird ignoriert und die Verzweigung in das Unterprogramm erfolgt nach dem errechneten jump-Ausdruck.

Beispiel:

```
0: ent N
1: gsb "X";jmpN
2: prt "end"
3: "X":end
4: prt "sub1";
 ret
5: prt "sub2";
 ret
6: prt "sub3";
 ret
```

Wird für N eine 3 eingegeben, verzweigt das Programm in das Unterprogramm in Zeile 4.

If-Befehl

Syntax:

`if` Ausdruck

Der if-Befehl wird zu Verzweigungen aufgrund logischer Entscheidungen verwendet. Nach einem if-Befehl wird der folgende Ausdruck ausgewertet. Ergibt der Vergleich 0 (falsch), läuft das Programm mit der nächsten Zeile weiter, falls davor nicht ein go to- oder go sub-Befehl steht. Ergibt die Auswertung einen Wert, wird die Aussage als wahr angesehen und das Programm läuft in der gleichen Zeile weiter. Der if-Befehl wird meist mit Vergleichsbefehlen oder Flags verwendet.

Beispiel:

```
0: ent A,B
1: if A=B;gto
 "one"
2: gto "zero"
3: "one":dsp
 "A=B"
4: stp
5: "zero":dsp
 "A#B"
6: end
```

Werte für A und B eingeben.
Wenn A=B, nach „One" springen.

Andernfalls zu „Zero" springen.
Bei Marke „One" A=B anzeigen.

Stop.

Bei Marke „Zero" A#B anzeigen.

Programm beendet.

Sind A und B gleich, wird A=B angezeigt, andernfalls wird A#B angezeigt.

Der if-Befehl kann auch mit anderen Befehlen, außer dem go to-Befehl im obigen Beispiel verwendet werden. Das Programm kann dann wie folgt gekürzt werden.

```
0: ent A,B
1: if A=B;dsp
 "A=B";stp
2: dsp "A#B"
3: end
```

Es ist zu beachten, daß kein go to-Befehl verwendet wird.

Verzweigungen in verschiedenen Richtungen

Der if-Befehl, zusammen mit einem go to- oder go sub-Befehl, ermöglicht die Verzweigung zu verschiedenen Stellen im Programm. Verzweigungen dieser Art erfolgen in der folgenden Form:

```
gto ... ; if ... ; gto ...
```

oder

```
gsb ... ; if ... ; gsb ...
```

Ist der if-Vergleich falsch, wird die Verzweigung durch den ersten go to- oder go sub-Befehl bestimmt. Ist der if-Vergleich wahr, bestimmt der zweite go to- oder go sub-Befehl die Verzweigung. Es können in einer Zeile go to- und go sub-Befehle verwendet werden.

Enthält eine Zeile mehr als einen go to oder go sub-Befehl, wird nur der zuletzt durchlaufene Befehl ausgeführt.

Beispiel:

```
20: gto 24;if
 X>30;gto 32;if
 X>40;gto "max"
21:
```

Ist X ⩽ 30, wird zu Zeile 24 verzweigt. Ist X > 30 und ⩽ 40, wird zu Zeile 32 verzweigt. Ist X > 40, wird zu der Zeile mit der Marke „max" verzweigt.

Dimensionierungsbefehl

Syntax:

`dim` Bezeichnung [, Bezeichnung , ...]

Bezeichnungen können sein: Einfache Variable.
Feld-Variable
(Bezeichnung [, Bezeichnung , ...])

Mit dem Dimensionierungsbefehl (dim) wird Speicherplatz für einfache und Feld-Variable reserviert und dabei den angegebenen Variablen der Wert 0 zugeordnet. In einem Dimensionierungsbefehl dürfen keine r-Variablen vorkommen, außer als Element.

In einem Dimensionierungsbefehl können die Elemente eines Feldes durch einen Ausdruck festgelegt werden.

Beispiel:

```
0: ent N,I,r2
1: dim A[N,I],
 B[r2],C[3,2*N]
```

Die Dimensionen werden mit Variablen festgelegt.

Variablen wird Speicherplatz in der entsprechenden Reihenfolge ihres Auftretens zugeordnet. Ist einer Variablen bereits Platz zugeordnet, erfolgt eine Fehlermeldung. Alle Variablen eines Dimensionierungsbefehls werden in einem zusammenhängenden Block im Speicher untergebracht. Dies ist für die Aufzeichnung von Daten wichtig.

Dimensionierungs-Befehle können an beliebiger Stelle in einem Programm stehen, sie dürfen jedoch nur einmal ausgeführt werden. Die Anzahl der Dimensionierungs-Befehle wird nur durch den Speicher begrenzt. Die Anzahl und die Größe der Elemente wird durch den Speicher und die Zeilenlänge begrenzt.

Beispiel:

```
0: dim N[2,2,2,
 2,2,2,2]
1: dim M[1000]
```

Reserviert 128 Feldelemente
Reserviert 1000 Feldelemente

Grenzwerte der Elemente

Der obere und der untere Wert der Elemente eines Feldes können angegeben werden. Zunächst muß die untere Grenze angegeben werden, die mit einem Doppelpunkt von der oberen Grenze getrennt wird.

Beispiel:

```
0: dim S[-3:0,
 4:6]
```

Reserviert 12 Feldelemente

Durch den folgenden Befehl wird gleich viel Speicherplatz reserviert:

```
0: dim X[4,3]
```

Reserviert 12 Feldelemente

Die Elemente des Feldes S werden wie folgt dargestellt:

S[−3,4]	S[−3,5]	S[−3,6]
S[−2,4]	S[−2,5]	S[−2,6]
S[−1,4]	S[−1,5]	S[−1,6]
S[0,4]	S[0,5]	S[0,6]

Ist die untere Grenze nicht angegeben wie in X(4,3), wird angenommen, daß sie wie in X [1:4, 1:3] gleich 1 ist.

Clear Simple Variables-Befehl

Syntax: `csv`

Mit dem Clear Simple Variables-Befehl (csv) werden alle zugeordneten einfachen Variablen von A bis Z auf 0 gesetzt. Mit diesem Befehl wird der zugewiesene Speicherplatz jedoch nicht gelöscht. Dadurch wird ein Fehler angezeigt, wenn die folgende Zeile ausgeführt wird:

```
0: 7→A:csv:dim A
```

Nicht zulässig, es kann nicht zweimal Platz für A reserviert werden.

List-Befehl

Syntax:

`list` [Anfang-Zeilennummer [, letzte gewünschte Zeilennummer]]

`list` Frei definierbare Funktionstaste

`list k`

Mit dem List-Befehl werden Programme, Teile von Programmen oder speziellen Funktionstasten zugeordnete Funktionen ausgedruckt. Steht der List-Befehl ohne weitere Angaben, wird das gesamte Programm ausgedruckt. Wird eine Zeilennummer angegeben, so wird das gesamte Programm ab dieser Zeile ausgedruckt. Werden zwei Zeilennummern angegeben, wird der Programmteil innerhalb dieser beiden Zeilennummern ausgedruckt. Um die Funktionen aller

spezieller Funktionstasten auszudrucken, wird List k eingegeben. Wird auf den List-Befehl eine spezielle Funktionstaste gedrückt, dann wird nur die Funktion dieser Taste ausgedruckt (nicht programmierbar). Der List-Befehl muß der letzte Befehl einer Zeile sein.

Beispiel:

`0: list`	Das gesamte Programm wird ausgedruckt.
`list 10,15`	Die Zeilen 10 bis 15 werden ausgedruckt.
`list4,4`	Zeile 4 wird ausgedruckt.
`list k`	Die Funktionen der frei definierbaren Funktionstasten werden ausgedruckt.
`list` [f10]	Die Funktion der speziellen Funktionstaste f10 wird ausgedruckt (nicht programmierbar).

Am Ende der Liste wird eine Prüfsumme mit ausgedruckt. Durch diese Prüfsumme können ausgetauschte oder weggelassene Zeilen und Buchstaben erkannt werden. Bei Unterschieden in Programmen wird auch eine unterschiedliche Prüfsumme ausgedruckt. Bei den folgenden zwei Programmen sind nur die Buchstaben „rt" in Zeile 1 vertauscht. Dadurch erscheinen zwei unterschiedliche Prüfsummen.

```
0: ent N
1: prt "satr.
 of",N
2: prt "is",rN
3: end
*23736
```

```
0: ent N
1: prt "sart.
 of",N
2: prt "is",rN
3: end
*23740
```

unterschiedliche Prüfsummen

Belegter und verbleibender Speicherplatz

Nachdem der Ausdruck erfolgt ist, werden zwei Werte angezeigt. Der erste Wert gibt die Gesamtlänge des Programms ohne Variable, Rücksprungadressen usw. in Byte an. Mit dem zweiten Wert wird der verbleibende Speicherplatz in Byte angegeben.

Beispiel:

```
468                                    6246
```

Programmlänge (in Byte) — Verbleibender Speicherplatz

ABSCHNITT 6
PROGRAMMKORREKTUR

Bei der Überarbeitung eines Programms können Fehler korrigiert oder das Programm auf den neuesten Stand gebracht werden. Dabei können Löschungen erfolgen, Zeilen eingesetzt oder geändert oder auch einzelne Zeichen eingesetzt oder geändert werden. Die zu ändernden Zeilen lassen sich einzeln in die Anzeige zurückholen. Mit der [STEP] Taste kann ein Programm Zeile für Zeile durchgetastet werden. In diesem Abschnitt werden die einzelnen Schritte zur Programmkorrektur beschrieben.

Auffinden von Fehlern

Der erste Schritt zur Korrektur ist, die Zeilen zu finden, die geändert werden sollen. Dies kann auf verschiedene Arten erfolgen. Eine Möglichkeit ist, das Programm mit der [STEP] Taste Zeile für Zeile durchzutasten. Nach jeder ausgeführten Zeile kann dann das Ergebnis überprüft werden.

Es können auch die Befehle Trace (trc), Stop (stp) und Normal (nor) verwendet werden. Werden Zeilen mit dem Trace-Befehl angezeigt, so werden alle Variable und Flags, die zugeordnet sind, ausgedruckt. Dadurch kann das Programm zeilenweise abgearbeitet werden. Mit dem Stop-Befehl kann das Programm bei jeder beliebigen Zeile gestoppt werden.

Mit dem Normal-Befehl können Trace und Stop wieder gelöscht werden.

Korrekturen

Der nächste Schritt ist die Korrektur. In vielen Fällen muß dabei nur ein Zeichen geändert werden. Es könnte jedoch auch möglich sein, einige Programmzeilen neu zu schreiben.

Um Zeichen innerhalb einer Zeile zu ändern, wird die Zeile durch Drücken der Taste [FETCH] und der Zeilen-Nummer aufgerufen. Durch Drücken der Taste [↵] wird diese Zeile angezeigt.

Soll die Änderung am Ende der Zeile vorgenommen werden, wird die Taste [BACK] gedrückt, soll die Änderung am Anfang der Zeile vorgenommen werden, wird die Taste [FWD] gedrückt. Steht das blinkende Zeichen über der zu ändernden Stelle, lassen sich einzelne Zeichen einsetzen, löschen oder überschreiben. Um Zeichen einzusetzen, wird die Taste [INS/RPL] gedrückt. Dadurch wird das blinkende Zeichen durch ein blinkendes ◀ ersetzt. Zeichen werden links von diesem Pfeil eingesetzt. Sollen Zeichen gelöscht werden, wird jedesmal [DELETE] gedrückt, um das Zeichen zu löschen. Um Zeichen zu ändern, muß dieses Zeichen ▒ in der Anzeige stehen (ist das Zeichen ◀ in der Anzeige, wird [INS/RPL] gedrückt, um ▒ anzuzeigen). Darauf werden die erforderlichen Zeichen eingegeben.

Um Zeilen innerhalb eines Programms zu ändern, werden sie mit den Tasten [FETCH], [↑] oder [↓] in die Anzeige geholt. Um diese Zeile zu löschen, wird die Taste [DELETE] gedrückt.

Ist die zu löschende Zeile mit einem relativen oder absoluten go to oder go sub-Befehl bezeichnet, so wird Fehler 36 angezeigt. Dann wird entweder der Delete-Befehl (del) mit dem Zusatzsternchen (Seite 92 ausgeführt, oder die Zeile mit dem entsprechenden go to oder go sub-Befehl der zu dieser Zeile geführt hat, geändert. Im folgenden Beispiel soll Zeile 25 gelöscht werden, aber Zeile 25 enthält einen go to-Befehl zu Zeile 27. Hier gibt es zwei Möglichkeiten.

Programmteil:

```
25: prt "number"
 ,N
26: if N=27;stp
27: N+1→N;gto 25
```

Alternative 1:

eingeben: `del 25,*`

drücken [R/S] löscht nur Zeile 25

Alternative 2:

Zeile 27 ändern in: `27: N+1→N;gto 26`

Zeile 25 in die Anzeige holen und die Taste [DELETE] Line drücken oder `del 25` ausführen.

Um eine Zeile einzufügen, wird die Zeile, vor die die neue Zeile gesetzt werden soll, angezeigt.

Darauf wird die neue Zeile in die Anzeige getippt und die Taste INSERT gedrückt, um sie zu speichern. Alle folgenden Zeilen werden dann automatisch umnumeriert (um 1 erhöht). Dabei werden auch alle go to oder go sub-Befehle automatisch umnumeriert. Im folgenden Beispiel wird eine Zeile zwischen den Zeilen 14 und 15 eingefügt.

```
14: prt "number
 of days";D
15: gto 19
```

Zuerst wird Zeile 15 angezeigt, dann wird die einzufügende Zeile eingetippt:

```
prt "number of weeks";W
```

Darauf wird die Taste line INSERT gedrückt.

Die Anzeige sieht jetzt wie folgt aus:

```
15►
```

Um zu überprüfen, daß die Zeile eingesetzt wurde, wird eingegeben: `list 14,16`

Das Programm sieht jetzt wie folgt aus:

```
14: prt "number
 of days";D
15: prt "number
 of weeks";W
16: gto 20
```

Dabei ist zu beachten, daß die Zeilennummer in dem go to-Befehl in Zeile 16 ebenfalls geändert wurde, da die alte Zeile 19 jetzt Zeile 20 ist.

Die Verzweigungsadresse eines jump-Befehls wird durch das Einfügen oder Löschen einer Zeile nicht beeinträchtigt.

Korrekturbefehle

Die Befehle Trace (trc), Stop (stp) und Normal (nor) können auch für Korrekturen verwendet werden. Diese drei Befehle können zwei verschiedene Bedeutungen haben, je nachdem, ob noch Parameter spezifiziert werden oder nicht.

Ausführung von Trace, Stop und Normal

Um diese Befehle richtig einsetzen zu können, muß man ihre Funktion verstehen. Es gibt eine „Master Flag", mit der sich die Betriebsarten Trace and Stop ein- und ausschalten lassen. Zusätzlich hat jede Zeile zwei Flags. Mit der Trace Flag läßt sich die Überwachung einer Zeile ein- und ausschalten. Mit der Stop-Flag läßt sich der Stop an einer Zeile ein- und ausschalten.

Diese Flags haben mit den Flags 0 bis 15, die in Abschnitt 5 beschrieben werden, nichts zu tun.

Trace-Befehl

Syntax:

trc [erste Zeilennummer (, Nummer der letzten Zeile)]

Mit dem Trace-Befehl (trc) wird die „Master Trace Flag" gesetzt. Werden in diesem Befehl Zeilennummern angegeben, so werden die „Trace Flags" an diese Zeilen gesetzt. Eine Zeilennummer spezifiziert nur diese eine Zeile, zwei Zeilennummern spezifizieren einen Programmblock von der ersten bis zur zweiten Zeilennummer.

Wird ein Trace bei einem laufenden Programm ausgeführt, so werden die Zeilennummer, die zugeordneten Variablen und der Zustand der Flags ausgedruckt.

Stop-Befehl

Syntax:

`stp` erste Zeilennummer [, zweite Zeilennummer]

Mit Stop-Befehlen (stp) können die „Master Trace Flag" und Stop Flags an den Anfang einer Zeile gesetzt werden. Läuft das Programm zu einer Zeile, bei der eine Stop Flag gesetzt ist (und wenn die Master Flag gesetzt ist), so hält das Programm am Anfang dieser Zeile an und die Zeilennummer wird angezeigt.

Durch Drücken der Taste CONTINUE oder STEP (siehe Beschreibung der Continue-und Stop-Tasten) läuft das Programm weiter.

Normal-Befehl

Syntax:

`nor` [erste Zeilennummer 1, [, zweite Zeilennummer 2]]

Mit dem Befehl Normal (nor) werden Stop und Trace Flags gelöscht.

Wird der Befehl nor ohne Zeilennummern ausgeführt, werden die Befehle Trace und Stop beendet. Einzeln gesetzte Stop und Trace Flags in den Programmzeilen werden jedoch nicht gelöscht. Folgt jedoch auf nor eine Zeilennummer, wird die Trace oder Stop Flag in dieser Zeile gelöscht und die Betriebsart Trace bleibt weiter bestehen. Folgen auf nor zwei Zeilennummern, so werden alle Trace und Stop Flags in dieser Zeilengruppe gelöscht.

Um alle Trace und Stop Flags innerhalb eines Programmes zu löschen, wird nor und die Nummer der ersten und der letzten Zeile des Programms eingegeben.

Beispiel:

Ein 100-Zeilen Programm enthält drei Teile mit schwierigen Operationen. Diese Teile können mit den folgenden Befehlen „getraced" werden:

```
trc5,15
trc40,50
trc70,85
```

Beim Abarbeiten des Programmes zeigt der Ausdruck auf Trace, daß Zeile 45 einen Fehler enthält. Diese Zeile wird dann korrigiert und nor ausgeführt, um die Master Trace Flag zu löschen. Das Programm wird wieder abgearbeitet, wobei aber jetzt die Zuordnungen nicht mehr ausgedruckt werden. Am Ende des Programmes wird klar, daß das Programm noch einen Fehler enthält. Jetzt werden nochmals die drei kritischen Teile durch Ausführen von trc ,,getraced''. Dadurch werden die Zeilen 5—15, 40—50 und 70—85 mit der Master Trace Flag markiert. Nach vollständiger Korrektur werden die Trace Flags der Zeilen mit nor 0,9999 gelöscht.

Normalerweise werden die Trace and Stop Flags durch einen record file Befehl nicht auf die Kassette aufgezeichnet. Sie können jedoch mit dem Zusatz (,,DB'') zum Befehl record file mit aufgezeichnet werden.

ABSCHNITT 7
ANWEISUNGEN

In diesem Abschnitt werden fünf Anweisungen behandelt. Diese Anweisungen unterscheiden sich von den Befehlen, da sie nur über die Tastatur ausgeführt werden können. Diese Anweisungen können nicht als Teil eines Programms gespeichert werden.

Run-Anweisung

Syntax:

`run` [Zeilennummer oder -Marke]

Mit der Anweisung Run werden alle Variablen, Flags und Unterprogramm-Rücksprungadressen gelöscht und darauf das Programm gestartet. Wird eine Zeilennummer angegeben, so wird das Programm beginnend mit dieser Zeile ausgeführt. Wird eine Marke angegeben, so beginnt die Programmausführung an der angegebenen Marke.

Beispiel:

`run` ↵	Ausführung des Programms von Zeile 0.
`run20` ↵	Programm wird ab Zeile 20 abgearbeitet.
`run "third"` ↵	Programm wird ab Marke „third" abgearbeitet.

Continue-Anweisung

Syntax:

`cont` [Zeilennummer oder -Marke]

Auf die Continue (cont) – Anweisung läuft das Programm weiter, ohne daß Variable, Flags oder Unterprogramm-Rücksprungadressen geändert werden. Ist keine Zeilennummer angegeben, läuft das Programm weiter, wo es gerade steht. Wird eine Zeilennummer angegeben, läuft das Programm ab dieser Zeile weiter. Ist nach einer Run- oder Continue-Anweisung eine Korrektur erfolgt oder ein Fehler aufgetreten, so wird das Programm beginnend mit Zeile 0 abgearbeitet.

Beispiel:

`cont` [ENTER] Programm läuft mit der angezeigten Zeilennummer weiter [CONTINUE]

`cont3` [ENTER] Programm läuft ab Zeile 3 weiter.

`cont"loop"` [ENTER] Programm läuft mit Marke „Loop" weiter.

Delete Line-Anweisung

Syntax:

`del` erste Zeilennummer [, zweite Zeilennummer] (, +)

Mit der Delete (del) – Anweisung können Zeilen oder Programmabschnitte gelöscht werden.

Wird nur eine Zeilennummer angegeben, so wird nur diese Zeile gelöscht. Werden zwei Zeilennummern angegeben, so werden alle Zeilen innerhalb dieses Blocks gelöscht. Um ein gesamtes Programm zu löschen und dabei die Variablen unverändert zu lassen, wird del Ø,9999 eingegeben.

Beispiel:

```
del 28 [ENTER]
```
Zeile 28 löschen.

```
del 13,20 [ENTER]
```
Zeilen 13 bis 20 löschen.

```
del 18,9999 [ENTER]
```
Programm von Zeile 18 bis zum Ende löschen.
(Die Variablen werden dabei nicht gelöscht).

Wird das (*) mit in die Anweisung aufgenommen, so werden go to oder go sub-Befehle, die sich auf die gelöschte Zeile beziehen, automatisch auf die nächstfolgende Zeile umnumeriert.

Beispiel:

```
22: ent U;if
 U=0;gto 24
23: U+T→T;C+1→C;
 gto 22
24: prt "Avg.
 Usage",T/C
25: prt "Total
 Usage",T
```

Eingeben: `del 24,*`

Drücken: [ENTER]

Drücken: [LIST] [2] [2] [,] [2] [4] [ENTER]

```
22: ent U;if
 U=0;gto 24
23: U+T→T;C+1→C;
 gto 22
24: prt "Total
 Usage",T
```

Erase-Anweisung

Syntax:

`erase` [`a` oder `v` oder `k` oder spezielle Funktionstaste]

Mit der Erase-Anweisung können Programme, Variable und die Funktionen von speziellen Funktionstasten wie folgt gelöscht werden:

Anweisung	Funktion
`erase`	Programme und Variable werden nach Ausführung dieses Befehls gelöscht.
`erase a`	Löscht den gesamten Speicher, wirkt wie kurzes Abschalten des Rechners.
`erase v`	Löscht alle Variablen.
`erase k`	Löscht alle Funktionen der speziellen Funktionstasten.
`erase` [fn]	Löscht die Funktion der angegebenen speziellen Funktionstaste.

(siehe Anhang D)

Fetch-Anweisung

Syntax:

`fetch` [Zeilennummer oder spezielle Funktionstaste]

Mit der Fetch-Anweisung werden einzelne Programmzeilen in die Anzeige geholt. Diese Möglichkeit ist besonders für Programmkorrekturen oder zur Überprüfung einzelner Zeilen wichtig. Wird die Funktion einer speziellen Funktionstaste aufgerufen, so wird die zugeordnete Funktion oder – wenn keine Funktion zugeordnet wurde – ein `f` mit der Nummer der Taste angezeigt. Wird nur die Anweisung Fetch eingegeben, so wird die Zeile 0 angezeigt.

Beispiel:

`fetch 10` Zeile 10 wird angezeigt.

`fetch` [fn] Die Funktion der Taste [fn] wird angezeigt.

ABSCHNITT 8
LIVE KEYBOARD

Mit der Möglichkeit, den Tischrechner auf „Live Keyboard" zu schalten, können mehrere Befehle oder auch Berechnungen ausgeführt werden, während ein Programm läuft. Das Programm läßt sich überwachen oder der Programmablauf kann geändert werden, während das Gerät auf „Live Keyboard" geschaltet ist. Mit den zwei in diesem Abschnitt beschriebenen Befehlen läßt sich die Betriebsart „Live Keyboard" ein- oder ausschalten.

So arbeitet Live Keyboard

Während ein Programm läuft, kann der Rechner in der Betriebsart „Live Keyboard" wie folgt über das Tastenfeld bedient werden.

- Die Eingabe erfolgt über das Tastenfeld, darauf wird die Taste [EXECUTE] gedrückt.
- Am Ende der momentanen Programmzeile wird die über das Tastenfeld eingegebene Zeile ausgeführt.
- Nach Ausführung dieser Zeile läuft das Programm weiter.

Live Keyboard-Arithmetik

In Betriebsart „Live Keyboard" können beliebige mathematische Operationen ausgeführt werden. Wird es erforderlich, bei laufendem Programm Berechnungen auszuführen, wird die Aufgabe eingetippt und die Taste [EXECUTE] gedrückt.

Befehle bei Betriebsart Live Keyboard

Mathematische Operationen sind nur ein kleiner Teil der Aufgaben, die bei Betriebsart „Live Keyboard" ausgeführt werden können. Soll z. B. das laufende Programm ausgedruckt werden, brauchen nur die Taste [LIST] und [EXECUTE] gedrückt zu werden.

Um eine Variable eines Programmes zu überprüfen, wird die Bezeichnung der Variablen eingegeben, wie z. B. A oder B [4] und [EXECUTE] gedrückt. Der Wert der Variablen wird dann angezeigt. Um den Wert einer Variablen über „Live Keyboard" zu ändern wird der neue Wert eingegeben. und der zu ändernden Variablen zugeordnet.

Um z. B. den Zähler `C+1→C` auf 0 zu setzen, wird `0→C` eingegeben und [EXECUTE] gedrückt.

Unterprogramme über Live Keyboard

Teile eines Programmes lassen sich über „Live Keyboard" als Unterprogramme mit dem gsb-Befehl ausführen. Mit dem folgenden Programmteil werden z. B. die im Programm verwendeten Variablen ausgedruckt:

```
11: "check":prt
 A;B;C;D;ret
```

Wird `gsb"check"` ausgeführt, so werden die Werte der Variablen ausgedruckt und das Programm läuft weiter.

Wird in einem solchen Unterprogramm ein return (ret), stop oder End-Befehl ausgeführt, so wird der „Live Keyboard"-Betrieb beendet und das Hauptprogramm läuft weiter. Ein zweites stop hält auch das Hauptprogramm an.

Frei definierbare Funktionstasten in Betriebsart Live Keyboard

Obwohl die frei definierbaren Funktionstasten f_0 bis f_{23} in Betriebsart „Live Keyboard" nicht definiert werden können, so lassen sie sich doch aufrufen und abarbeiten. In diesem Beispiel wird der Ablauf eines Programmes mit den speziellen Funktionstasten geändert.

Die frei definierbaren Funktionstasten sind wie folgt definiert:

[f0]	[f1]	[f2]
`*1→F`	`*2→F`	`*3→F`

Das Programm sieht wie folgt aus:

```
0: "Wait":dsp
 "waiting";wait
 100;jmp F
1: gto "first"
2: gto "second"
3: gto "third"
4: "first":prt
 "first";0→F;
 gto "Wait"
5: "second":prt
 "second";0→F;
 gto "Wait"
6: "third":prt
 "third";0→F;
 gto "Wait"
```

Beim Programmablauf wird so lange `waiting` angezeigt, bis eine der speziellen Funktionstasten gedrückt wird. Darauf verzweigt das Programm zu einer Zeile, in der entweder `first`, `second` oder `third` ausgedruckt wird. Obwohl dies ein einfaches Beispiel ist, zeigt es doch, wie der Programmablauf in Betriebsart „Live Keyboard" geändert werden kann.

Stop-Taste in Betriebsart Live Keyboard

Drücken der Taste (STOP) beendet die Betriebsart „Live Keyboard". Bei zweimaligem Drücken wird auch das Programm angehalten.

Einschränkungen bei Betriebsart Live Keyboard

Operationen, die das gespeicherte Programm oder die speziellen Funktionstasten verändern würden und Operationen, die die Ausführung eines Programmes direkt beeinflussen, sind nicht erlaubt. Es sind dies folgende Anweisungen:

Anweisungen	Fehler
run	error 03
cont	error 03
fetch	error 03
erase	error 03
del	error 03
Befehle:	
ent	error 13
end	error 09
gto (in einem Live Keyboard Unterprogramm erlaubt)	error 09
ldp	error 64
ldk	error 64
ldf (Programm File)	error 64

Bei den folgenden Tasten erfolgt ein akustischer Signal in Live Keyboard und die Funktion wird nicht ausgeführt.

Anzeige

Wird jedoch die Anzeige innerhalb des Programms fortlaufend verwendet, so erfolgt keine Anzeige der über die Tastatur eingegebenen Werte, auch wenn RECALL gedrückt wird. Durch Drücken von ↑ oder ↓ wird die Zeile für etwa eine Sekunde angezeigt. Die Anzeige bleibt jedoch stehen und das Programm wird so lange unterbrochen, solange eine dieser Tasten gedrückt wird. Läuft z. B. das folgende Programm im Rechner:

```
0: dsp "Live
 Keyboard";wait
 100
1: gto 0
```

und die folgende Zeile wird über „Live Keyboard" eingegeben, so wird die Zeile nicht angezeigt: `prt r25÷A`

Drücken: ↑ oder ↑ . Dadurch wird die Zeile etwa eine Sekunde lang angezeigt.

Wird jetzt die Taste ENTER gedrückt, so wird die Zeile ausgeführt und 5 wird in A gespeichert und ausgedruckt.

Ergebnisse von Berechnungen, die über „Live Keyboard" ausgeführt werden, verschwinden wieder, wenn das laufende Programm die Anzeige verwendet. Durch Drücken der Tasten ↑ oder ↑ wird nur die eingegebene Zeile angezeigt und nicht das Ergebnis. Das Ergebnis läßt sich anzeigen, indem am Ende der Zeile ein Wait Befehl eingefügt wird. (Z. B. 10+12; wait 1000). Einer frei definierbaren Funktionstaste kann eine Funktion zugeordnet werden, um die Anzeige so lange stehen zu lassen, daß man sie ablesen kann, wie im folgenden Beispiel:

Drücken: FETCH f0

Eintippen: `*;wait 1000`

Drücken: STORE

Wird jetzt eine Aufgabe eingegeben wie `5*6` und f0 an Stelle der ENTER Taste gedrückt, so bleibt das Ergebnis etwa eine Sekunde in der Anzeige stehen.

Einschalten der Betriebsart Live Keyboard

Syntax: `lke`

Mit dem Befehl „Live Keyboard Enable" (lke) wird die Betriebsart „Live Keyboard" eingeschaltet.

Beispiel:

```
31: lke
```
Live Keyboard eingeschaltet.

Nach dem Einschalten und nach Drücken der Taste RESET ist der Rechner immer in Betriebsart „Live Keyboard". Mit dem Befehl „Live Keyboard Disable" (lkd) wird die Betriebsart „Live Keyboard" ausgeschaltet.

Befehl zum Abschalten der Betriebsart Live Keyboard

Syntax: `lkd`

Mit dem Befehl „Live Keyboard Disable" (lkd) wird die Betriebsart „Live Keyboard" abgeschaltet.

Beispiel:

```
0: lkd
```

Die erste Zeile dieses Programms schaltet die Betriebsart „Live Keyboard" ab.

Dieser Befehl wird verwendet, wenn ein laufendes Programm nicht unterbrochen werden soll. Dabei kann über einen „Live Keyboard Enable"-Befehl (lke) innerhalb eines Programms „Live Keyboard" wieder eingeschaltet werden.

STOP , REWIND und RESET sind die einzigen Tasten, die während eines laufenden Programms ausgeführt werden, wenn „Live Keyboard" abgeschaltet ist.

Werden Operationen mit der Kassette ausgeführt, so wird nur RESET akzeptiert.

ABSCHNITT 9
MAGNETBANDKASSETTE

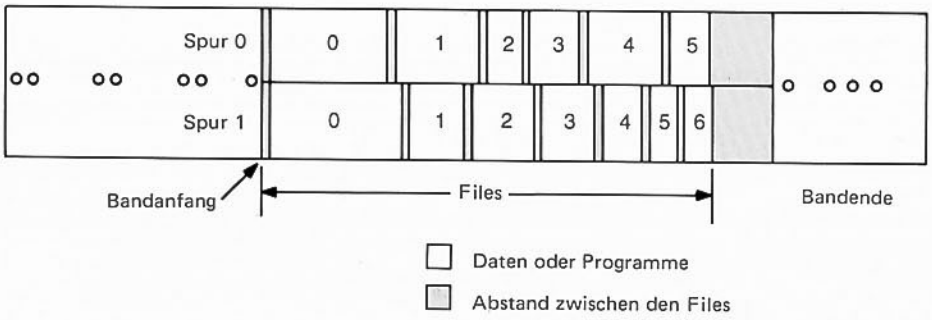
Die mit dem Tischrechner 9825A verwendete Magnetbandkassette hat eine sehr hohe Speicherdichte. Die Speicherung von Programmen und Daten, die Pflege und das Einsetzen der Magnetbandkassette werden in diesem Abschnitt beschrieben.

Technische Daten

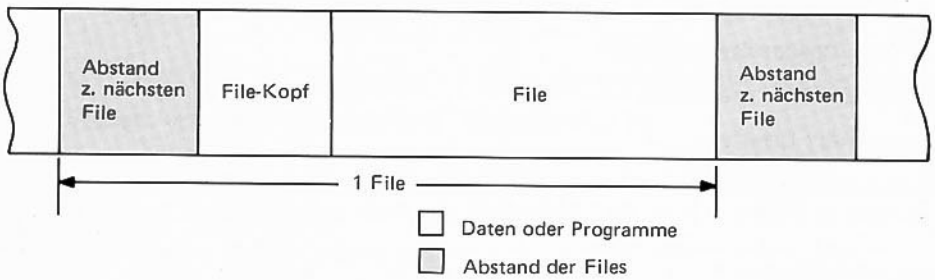
Typische Datenübertragungsgeschwindigkeit (Geschwindigkeit, mit der Informationen auf der Kassette gespeichert oder von der Kassette eingelesen werden können)	2750 Byte/Sek.
Typische Zugriffsgeschwindigkeit (Geschwindigkeit, mit der das Band am Magnetkopf vorbeigeführt wird, wenn ein File gesucht wird)	14300 Byte/Sek.
Typische Rückspulgeschwindigkeit (Von Anfang bis Ende)	19 Sekunden
Typische Löschzeit (Eine vollständige Spur)	40 Sekunden
Normale Bandlänge	42,67 Meter (140ft)
Spuren	2

Einteilung des Magnetbandes

So wird das Magnetband eingeteilt:



Eine einzelne File ist wie folgt eingeteilt:



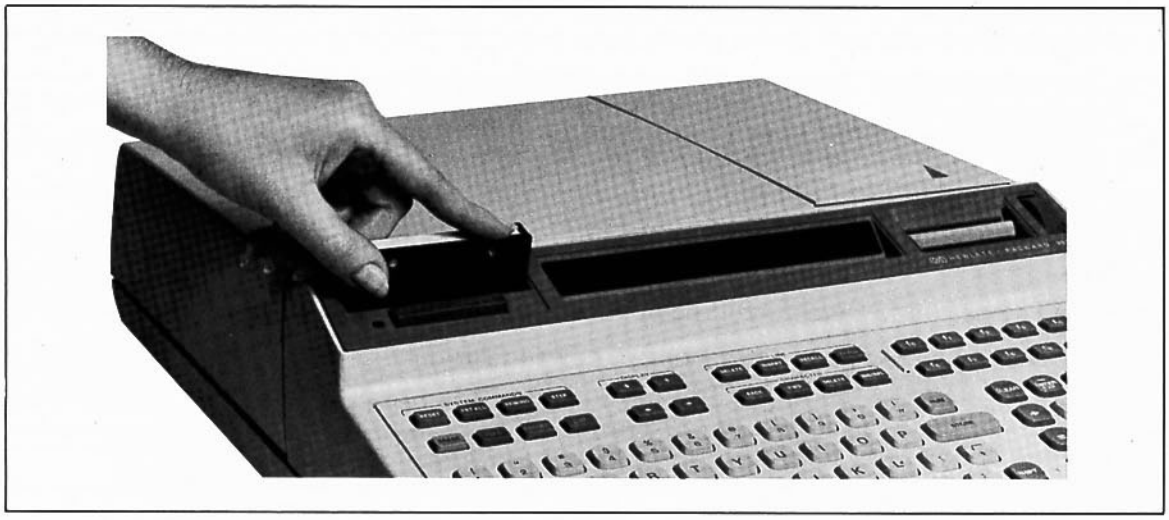
Bandkassette

Die unten gezeigte Bandkassette dient zum Speichern von Programmen, Daten und den Funktionen der frei definierbaren Funktionstasten.

Zur Aufzeichnung muß der Schieber mit der Aufschrift RECORD nach rechts geschoben werden.

Einsetzen der Kassette

Die Kassette wird so eingesetzt, daß die beschriftete Seite nach hinten (zum Rechner) zeigt.



Einsetzen der Kassette

Pflege der Kassetten

Reinigen des Magnetkopfes und des Antriebsrades

Die meisten bei Kassetten auftretenden Fehler sind auf Schmutz und Staub zurückzuführen. Durch einige grundlegende Vorsichtsmaßnahmen lassen sich diese Fehler sehr stark reduzieren.

- Der Magnetkopf und das Antriebsrad sind wenigstens nach 8 Stunden Betriebszeit zu reinigen; in staubiger Umgebung jedoch öfter.
- Die Bandkassette muß immer zurückgespult werden, bevor sie herausgenommen wird.
- Die Abdeckklappe der Bandkassette muß immer sauber sein.
- Die Kassette ist immer in den Plastikbehälter zurückzulegen.

Zwei weitere Faktoren können die Zuverlässigkeit der Kassette beeinflussen.

Auf Band gespeicherte Daten und Programme können durch starke Magnetfelder gelöscht werden. Durch Beschädigung des Bandes, d. h. wenn das Band gefaltet ist, können Probleme beim Aufzeichnen oder Einladen auftreten. Für alle kritischen Programme oder Daten sollte eine zweite Bandkassette angelegt werden.

Anhang F enthält Anweisungen, um Fehler bei der Aufzeichnung zu korrigieren.

Rückspulbefehl

Syntax: `rew`

Mit dem Rewind-Befehl (rew) wird das Band auf den Anfang zurückgespult. Dieser Befehl hat die gleiche Funktion wie die Rückspultaste [REWIND]

Das Rückspulen des Bandes ist ein selbständiger Vorgang, d. h. andere Operationen des Rechners werden dabei nicht beeinträchtigt.

Spurbefehl

Syntax:

`trk` Spur – Nummer

Mit dem Track-Befehl (trk) wird die Spur 0 oder die Spur 1 eingestellt. Nach diesem Befehl werden alle weiteren Operationen mit der eingestellten Spur ausgeführt. Wird der Rechner eingeschaltet, wird RESET gedrückt oder der Befehl „Erase a" ausgeführt, ist automatisch Spur 0 eingeschaltet. Diese Spur bleibt eingeschaltet, auch wenn die Kassette herausgenommen oder RUN gedrückt wird.

Die Nummern der Spuren werden nur mit 0 oder 1 bezeichnet.

HINWEIS

Wird RESET gedrückt, der Befehl `erase 0` ausgeführt oder der Rechner eingeschaltet, wird automatisch auf Spur 0 geschaltet. Wenn nicht durch einen Befehl auf Spur 1 umgeschaltet wird, erfolgen alle Operationen nur mit Spur 0. Falls dies nicht beachtet wird, können wichtige Programme oder Daten verlorengehen.

Identify File-Befehl

Syntax:

`idf` [File-Nummer [, File-Typ [, belegte File-Größe [, absolute File-Größe [, Spur-Nummer]]]]]

Mit dem Identify File-Befehl (idf) können die Parameter der nächstfolgenden File identifiziert werden. Allen fünf Abfrage-Variablen können dabei Werte zugeordnet werden. D. h. wenn der Befehl ausgeführt wird, werden allen angegebenen Variablen Werte zugeordnet. Wird nur eine Variable spezifiziert wie z. B.: idf A, so wird nur die File Nummer angegeben. Um den File-Typ zu ermitteln, müssen zwei Variable angegeben werden; drei Variable sind erforderlich, um die File-Größe in Byte zu ermitteln. Mit vier Variablen wird die absolute File-Größe in Byte ermittelt, zur Abfrage der Spur-Nummer sind fünf Variable erforderlich.

Die File-Typen sind wie folgt eingeteilt:

0	Null-File (mit der absoluten Grösse 0)
1	Binär-Programm
2	Numerische Werte
3	Zeichenketten oder Zeichenketten und numerische Werte (Zeichenketten-ROM erforderlich)
4	Speicher-File (durch rcm-Befehl)
5	Tasten-File
6	Programm-File

Wird eine Bandkassette eingesetzt, wird die Spur geändert oder [RESET] gedrückt oder der Befehl `erase a` ausgeführt, ist die Position des Bandes unbekannt. Ist die Stellung des Bandes nicht bekannt, wie nach der Umschaltung der Spuren, so muß mindestens eine Variable angegeben werden, weil sonst Fehler 45 angezeigt wird. Ist der Befehl Identify-File ausgeführt, wird das Band wieder automatisch an den Anfang der File zurückgespult.

Beispiel:

`idf A,B,C,D,E`	Die augenblickliche File identifizieren und die File-Nummer, den File-Typ, die File-Größe und die absolute File-Größe sowie die Spur-Nummer den Variablen A, B, C, D und E zuordnen.
`idf A,A,A`	Die Filegröße A zuordnen.

Find File-Befehl

Syntax:

`fdf` [File-Nummer]

Mit dem „Find File"-Befehl (fdf) wird die angegebene File auf der gerade eingestellten Spur der Bandkassette gesucht.

Das Band wird dann an den Anfang dieser File positioniert. Die File-Nummer kann auch ein Ausdruck sein. Bei einem Find File-Befehl ohne Parameter, wird File 0 gesucht.

HINWEIS

Wird eine File-Nummer angegeben, die nicht existiert, so wird durch den nächsten Befehl für die Bandkassette (außer Find File (fdf) oder Rewind (rew)) Fehler 65 angezeigt.

Wie das Rückspulen des Bandes, ist der Find File-Vorgang unabhängig von anderen Funktionen und kann jederzeit parallel ausgeführt werden.

Beispiel:

```
fdf8
4:fdf A[3]
```

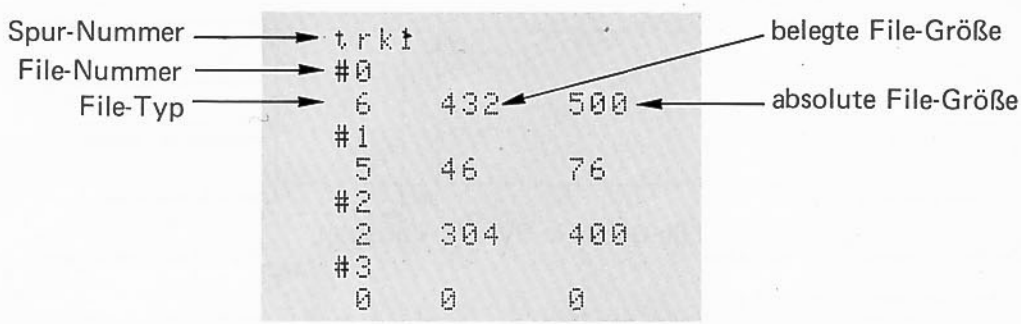
File 8 suchen.

Die durch den Wert von A (3) angegebene File suchen.

Tape List-Befehl

Syntax: `tlist`

Mit dem Tape List-Befehl (tlist) werden die Files einer Bandkassette identifiziert. Anfangend bei der augenblicklichen Stellung des Bandes werden die Spur, die File-Nummer, der File-Typ, die File-Größe in Byte und die absolute File-Größe wie im folgenden Beispiel ausgedruckt.




```
trk
#0
 6    432    500
#1
 5    46     76
#2
 2    304    400
#3
 0    0      0
```

Die File-Typen werden wie folgt eingeteilt:

- Ø Null-File (absolute Grösse Ø)
- 1 Binär-Programm
- 2 Numerische Daten
- 3 Zeichenketten oder Zeichenketten und numerische Daten (das Zeichenketten-ROM muß eingesetzt werden, weil sonst beim Laden dieses File-Typs Fehler SØ angezeigt wird.
- 4 Speicher-File (durch rcm-Befehl)
- 5 Tasten-File
- 6 Programm-File

Wird während des tlist-Befehls die Stop-Taste gedrückt, so wird dieser Befehl beendet. Normalerweise ist der Befehl beendet, wenn die Ø-File erreicht ist.

Eine sehr einfache Art zur Feststellung, welche Spur eingeschaltet ist, ist den Befehl „tlist" einzugeben und (STOP) zu drücken.

Es kann auch der Identity File Befehl wie folgt verwendet werden: `idf T;T;T;T;T;T` (ENTER)

Markieren von Bändern

Mark-Befehl

Syntax:

`mrk` Anzahl der Files[,File-Größe in Byte [, Variable]]

Mit dem Mark-Befehl (mrk) wird das Band in verschiedene Bereiche (Files) eingeteilt. Dabei wird ein File mehr als die angegebene Anzahl von Files markiert. Diese File ist die Ø-File. Sie ist der Anfangspunkt zur Markierung weiterer Files. Die Ø-File hat die absolute Größe von Ø. Die File-Größe wird in Byte angegeben. Wird eine ungerade Zahl von Byte angegeben, wird automatisch ein Byte mehr markiert. Werden z. B. 111 Byte angegeben, so werden 112 Byte markiert.

Um Files markieren zu können, muß die Position des Bandes bekannt sein. Ist die Position nicht bekannt, wird ein Find File (fdf) – oder Rewind (rew) – Befehl ausgeführt, um das Band richtig zu positionieren.

Der Mark-Befehl, bei dem die ersten zwei Parameter Ø sind, ist eine Besonderheit, er wird im Anfang F beschrieben.

Die Anzahl der Files und die File-Größe können beide Ausdrücke sein. Wird eine Variable angegeben, so wird die File-Nummer der letzten verfügbaren File zugewiesen. Ist der Wert der Variablen positiv, so sind alle angegebenen Files markiert. Ist der Wert negativ, so war das Band zu Ende, ehe alle angegebenen Files markiert werden konnten. In jedem Fall stellt der absolute Wert der Variablen die letzte verwendbare File dar.

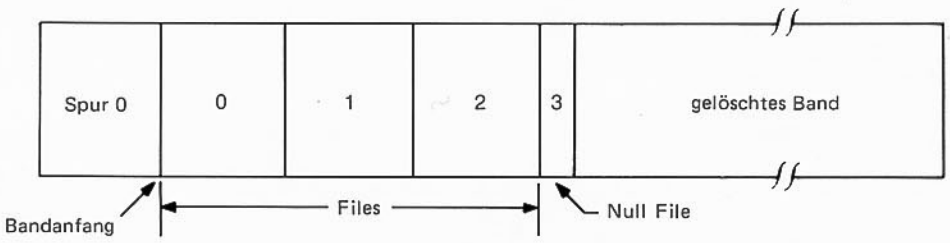
Beispiel:

Die Spur Ø soll mit drei Files mit einer Länge von je 320 Byte markiert werden. Dies wird mit dem folgenden kurzen Programm erreicht.

```
0: rew
1: trk 0
2: mrk 3,320,X
3: ert X+1
4: end
```

Kassette zurückspulen.
Auf Spur Ø schalten.
Drei Files mit je 320 Byte markieren.
Den Rest von Spur Ø löschen.
Ende des Programms.

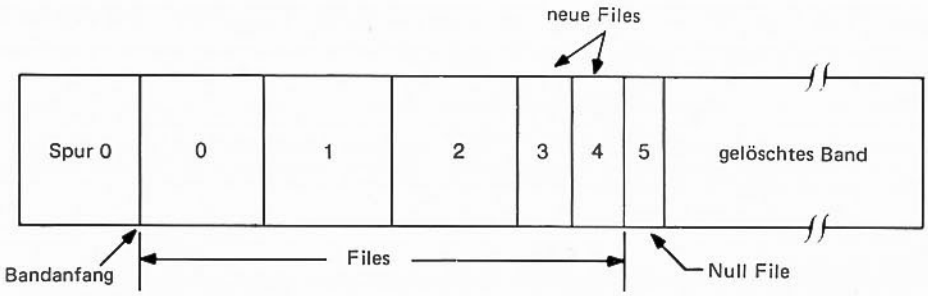
Das Band wird auf den Anfang von File 3 gesetzt. Es ist dann wie folgt markiert:



Darauf sollen zwei Files mit einer Länge von je 80 Byte markiert werden.

Es wird folgender Befehl verwendet: `mrk2,80`

Das Band sieht jetzt wie folgt aus:



Um 2 Files mit je 300 Byte ab File 4 zu markieren wird folgendes eingegeben:

```
fdf4;mrk2,300
```

Bestimmung der Größe zum Markieren einer File

Um ein File für ein im Rechner abgespeichertes Programm zu markieren, wird die Anweisung „list – 1" eingegeben und ausgeführt. Die Zahl auf der linken Seite der Anzeige gibt genau die zur Abspeicherung erforderliche Anzahl der Byte an. Es empfiehlt sich, die File etwas größer zu markieren, so daß auch spätere Programmänderungen, durch die das Programm größer werden kann, mit in der File abgespeichert werden können.

Bei Daten-Files sind je 8 Byte für ein Datenelement erforderlich. Um z. B. Daten, die unter den Variablen A und B gespeichert sind, zu registrieren, muß die File 16 Byte lang sein.

Bei Files für die speziellen Funktionstasten sind 1 Byte für jedes gespeicherte Zeichen unter der Taste plus 2 Byte für jede definierte Taste erforderlich. Ist die Anzahl der Byte für die Taste ungerade, wird ein Byte hinzuaddiert. Diese Summe ist dann die Mindestgröße für das File.

Für eine File zur Abspeicherung des Speicherinhalts des Rechners (mit rcm-Befehl) wird die File für die Größe des gesamten Speichers des Rechners markiert:

8192 Byte für die Standardausführung
16384 Byte für Option 001
24576 Byte für Option 002
32766 Byte für Option 003

Typische Speicherkapazität

File Größe (Byte)	Typische Anzahl von Files pro Spur	Byte pro Spur
50	827	41350
100	656	65600
250	404	101100
500	239	119500
750	170	127500
1000	131	131000
2500	56	140000
5000	28	140000
7500	19	142500
10000	14	140000

Durch die interne Verwaltung der Files ist die Anzahl der Byte/Spur bei unterschiedlichen File-Größen verschieden.

Berechnung der Bandkapazität

Die Anzahl der Files, die auf der Bandkassette gespeichert werden können, hängt von der Größe der einzelnen Files ab. Nach den folgenden Formeln kann die Anzahl der Files, die auf Band gespeichert werden können, berechnet werden.

$L = 1.278 + .209 \text{ int } (A/256 + .999) + .0105A$

Dabei sind:

A = absolute File-Größe in Byte.
L = Länge der File in Zoll.

a) Die typische Kapazität pro Spur:
   Anzahl der Files pro Spur = int (1665/L)

b) Die Minimum-Kapazität pro Spur:
   Anzahl der Files pro Spur = int (1498/L)

Mit dem folgenden Programm können mehr Files markiert werden und es läßt sich berechnen, wieviel Prozent einer Spur schon verwendet sind.

```
0: rew;fxd 0
1: ent "Track,
 0 or 1?",T;trk
 T
2: ent "New tape
 ? Yes=1 No=0",N
3: if N;gto "Mar
 k"
4: 0→F→L
5: fdf F;idf A,
 A,A,A
6: if A=0;gto
 "Mark"
7: gsb "L"
8: Q+L→L;F+1→F;
 gto 5
9: "Mark":prt
 "% of Tape"
10: prt "   Mark
 ed",L/1665*100→
 r0
11: if r0>100;
 prt "All Marked
 ";gto "out"
12: ent "Mark
 more files?
 Yes=1 No=0",M
13: if M=0;gto
 "out"
14: ent "Length
 of files",A;
 gsb "L"
15: dsp "Number
 of files",A;
 "long?"
16: ent "",I;I*
 Q+L→r1
17: if (r1/1665→
 Z)>=1;prt "Too
 long...","That
 is %",Z*100;
 gto "out"
18: r1→L;mrk I,
 A,R;if R<0;dsp
 "100% Marked";
 gto "out"
19: gto "Mark"
20: "out":dsp
 "That's it";end
21: "L":1.278+
 .209int(A/256+
 .999)+.0105A→Q;
 ret
```

Markieren von neuen Bändern

Da ein neues Band noch nicht in Files eingeteilt ist, muß es vor der Markierung zurückgespult werden. Um z. B. 4 Files zu je 200 Byte zu markieren, werden folgende Befehle ausgeführt:

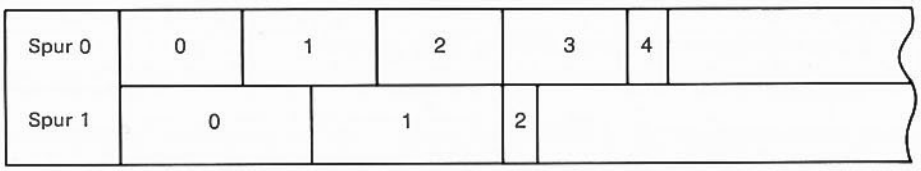
```
rew
mrk4,200
```

Markieren von schon verwendeten Bändern

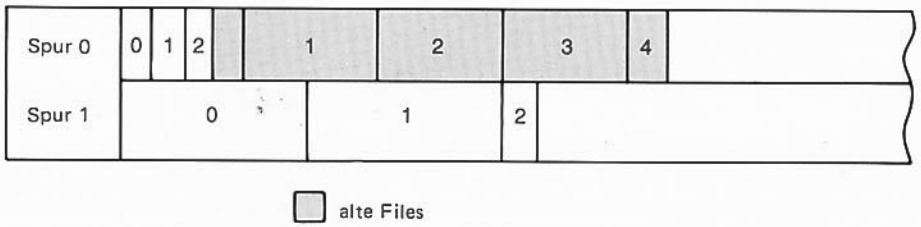
Wird ein Band ummarkiert, so könnte es passieren, daß einige alte Files erhalten bleiben. Diese Files könnten eingelesen werden, wenn fälschlicherweise die Spur geändert wird.

Beispiel:

Spur Ø besteht aus 4 Files zu je 1000 Byte und Spur 1 aus 2 Files zu je 1500 Byte.

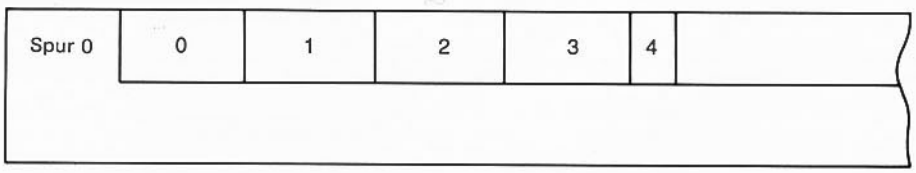


Dann wird Spur Ø von Bandanfang an umnumeriert in 2 Files zu je 200 Byte.

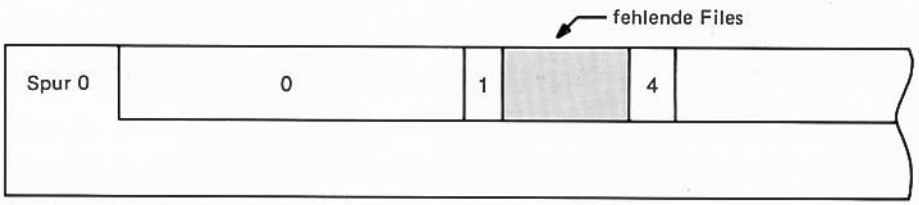


Ist das Band auf Spur 1, File 1 positioniert und wird `trkØ` ausgeführt, so steht das Band auf der alten File 1 der Spur Ø.

Anstelle dieser doppelten Files können auch Files fehlen. Nehmen wir an, daß Spur Ø mit 4 Files zu je 1000 Byte markiert ist:



Wird das Band jetzt zurückgespult und mrk 1,3000 ausgeführt, so würden Files fehlen:



Um alte Files zu löschen, wird der Befehl erase tape (ert) ausgeführt.

`rew`	Rückspulen des Bandes (Spur Ø)
`mrk2,200,A`	2 Files zu je 200 Byte markieren
`ertA+1`	Band ab File Ø löschen

HINWEIS

Beim Neumarkieren von Bändern müssen alte Files mit dem erase tape Befehl (ert) gelöscht werden.

Erase Tape-Befehl

Syntax:

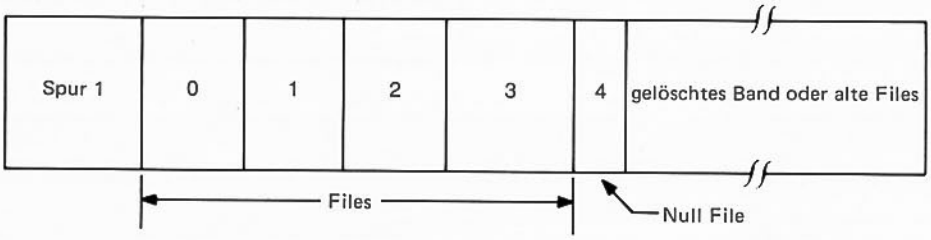
ert File-Nummer

Ist das Band markiert, so kann die eingeschaltete Spur mit dem erase tape-Befehl (ert) ab der angegebenen File-Nummer gelöscht werden. Nach der Löschung wird das Band auf die angegebene File-Nummer gestellt und eine File Null wird markiert. Die File Null gilt als Anfangspunkt zur Markierung weiterer Files (s. Markierungsbefehl).

Die File-Nummer kann auch ein Ausdruck sein.

Beispiel:

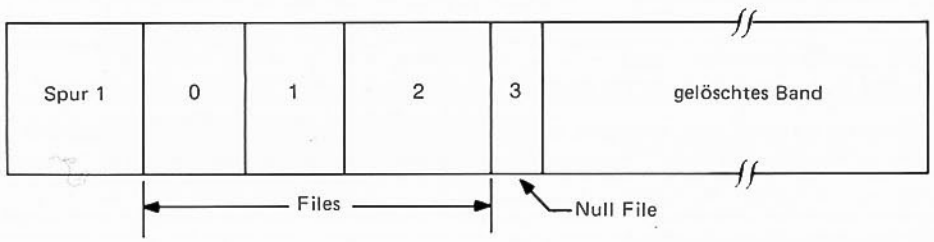
Ein Band hat auf Spur 1 die folgende Struktur:



Mit dem folgenden Programm werden alle Files ab File 3 auf Spur 1 bis zum Ende gelöscht.

```
0: trk 1
1: ert 3
2: end
```

Nach diesem Programm sieht die Struktur des Bandes wie folgt aus:



Spur Ø ist nicht verändert.

Record File-Befehl

Mit dem Record File-Befehl (rcf) können Daten und Programme gespeichert werden.

Aufzeichnen von Programmen

Syntax:

```
rcf     [File-Nummer [, Nummer der ersten Zeile [, Nummer der letzten Zeile]]
        [, "SE" oder "DB" ]]
```

Programme oder Teile von Programmen werden mit dem Record File-Befehl (rcf) gespeichert. Ist keine File-Nummer angegeben, so wird automatisch die File-Nummer Ø eingesetzt. Werden keine Zeilennummern angegeben, so wird das gesamte Programm in der angegebenen File gespeichert. Wird eine Zeilennummer angegeben, so wird das Programm ab dieser Zeilennummer bis zum Ende gespeichert.

Werden zwei Zeilennummern angegeben, so wird der dazwischenliegende Programmteil einschließlich dieser beiden angegebenen Zeilen gespeichert.

Die File-Nummer und die Bezeichnung der letzten Zeile können Konstanten, Variable oder Ausdrücke sein. Die erste Zeile kann eine Konstante oder ein Ausdruck wie 1A sein, jedoch keine Variable. Wird eine Variable, wie rcf1, A verwendet, so wird der Wert von A als numerischer Wert aufgezeichnet. Um ein Programm, ab der Zeile, die durch A gekennzeichnet ist, aufzuzeichnen, so wird rcf1, 1A eingegeben. Steht ein „SE" (für geschützt) hinter dem Befehl, so ist das gespeicherte Programm geschützt. Wird ein geschütztes Programm in den Rechner eingeladen, so kann es weder ausgedruckt noch angezeigt werden. Es läßt sich jedoch wieder auf eine Bandkassette aufnehmen.

Steht „DB" hinter dem Befehl, so werden Trace oder Stop Flags mit dem Programm gespeichert (Seite 88).

Bevor ein Programm auf Band aufgenommen werden kann, muß eine File markiert werden. Diese File muß mindestens so groß sein wie das aufzunehmende Programm (s. Mark-Befehl)

Beispiel:

```
7: rcf 8,3
```

Das Programm beginnend mit Zeile 3 bis zum Ende in File 8 speichern.

Aufzeichnen von Daten

Syntax:

`rcf` File-Nummer, Daten-Liste

Wird diese Syntax verwendet, so werden mit dem Record File-Befehl (rcf) Daten gespeichert. Die Liste kann aus einfachen Variablen, Feld-Variablen oder r-Variablen bestehen. r-Variable können jedoch nicht mit einfachen und Feld-Variablen im selben Befehl gemischt werden. Die File-Nummer kann auch ein Ausdruck sein.

Um ein gesamtes Feld zu speichern, muß hinter der Bezeichnung ein Sternchen in Klammern stehen. Beispiel:

```
rcf 2,S[*]
```

bezieht sich auf das gesamte Feld S in File 2

Einfache und Feld-Variable müssen im Speicher des Rechners hintereinander stehen, d. h. daß sie in der Datenliste in der Reihenfolge ihrer Zuordnung stehen müssen. Stehen die Variablen in einem Dimensionierungsbefehl (dim), dann müssen sie in der gleichen Reihenfolge auch in dem rcf-Befehl stehen.

Beispiel:

```
0: dim A[10,10]
```

Dem Feld A werden 100 Elemente zugeordnet (800 Byte).

```
1: 0→X
```

Der Variablen X ist eine Variable zugeordnet (8 Byte).

```
2: X+1→X
```

Verändert nicht den X zugeordneten Speicherplatz.

```
3: 1→I
```

Der Variablen ist der Wert 1 zugeordnet (8 Byte).

```
4: rcf 5,A[*],X,
I
```

Das Feld A und die Variable X werden in der gleichen Reihenfolge der Zuordnung in File 5 gespeichert (insgesamt 102 Ziffern oder 816 Byte).

Ist eine r-Variable in der Datenliste angegeben, so werden alle r-Variablen von rØ bis zu dieser r-Variablen aufgezeichnet. Werden zwei r-Variable angegeben, so werden alle r-Variablen von der ersten bis zur zweiten r-Variablen aufgezeichnet.

Hinweise zur Aufzeichnung von Daten

Werden Daten auf der Kassette gespeichert, so muß die Variablenliste in der gleichen Reihenfolge erscheinen wie sie im Speicher abgespeichert ist.

Beispiel:

```
0: ent A
1: 2*A→B
2: dim C,X,Y,Z
   .
   .
   .
15: rcf A,B,C,X,
Y,Z
```

In diesem Programm stehen die Variablen A und B nicht im Dimensionierungsbefehl. Die Variablen C, X, Y und Z stehen in einem Dimensionierungsbefehl.

Wird jetzt innerhalb des Programmes B vor A zugeordnet, so wird bei Zeile 15 Fehler 56 angezeigt, da die Variablen in der Reihenfolge ihrer Zuordnung aufgeführt werden müssen. Da die einzelnen Zeilen nicht notwendigerweise in ihrer numerischen Reihenfolge abgearbeitet werden, ist es manchmal schwierig, die Reihenfolge zu bestimmen, in der die Variablen zugeordnet sind. Sollen Variable in der File aufgezeichnet werden, wird empfohlen, sie in einem Dimensionierungsbefehl unterzubringen.

Load Program-Befehl

Syntax:

`ldp` [File-Nummer [, 1. Zeilennummer [, 2. Zeilennummer]]]

Mit dem Load Program-Befehl (ldp) werden Programme aus der angegebenen File in den Rechner eingeladen und automatisch abgearbeitet. Dabei werden alle Variable, alle Unterprogramm-Rücksprungadressen und alle Flags gelöscht.

Wird nur eine File-Nummer angegeben, so wird das Programm dieser File, beginnend mit Zeile Ø eingeladen und auch von Zeile Ø beginnend automatisch abgearbeitet. Wird eine File-Nummer und eine Zeilennummer angegeben, so wird das Programm ab dieser Zeile eingeladen und auch ab dieser Zeile abgearbeitet. Werden alle drei Parameter angegeben, so wird das Programm der angegebenen File-Nummer beginnend mit der ersten Zeilennummer eingeladen, es wird dann jedoch beginnend mit der zweiten Zeilennummer abgearbeitet. Werden keine Parameter angegeben, so werden sie alle als Ø angesehen. Alle drei Parameter können auch Ausdrücke sein. Der Load Programm-Befehl kann nur als letzter Befehl in einer Zeile stehen.

Dieser Befehl ist in Betriebsart „Live Keyboard" oder in einem Enter-Befehl nicht zulässig.

Beispiel:

`ldp2`	Lädt das Programm ab Zeile Ø aus File 2 ein und arbeitet es ab Zeile Ø ab.
`ldp8,2`	Lädt das Programm beginnend mit Zeile 2 aus File 8 ein und arbeitet es ab Zeile 2 ab.
`ldp16,3,0`	Lädt das Programm beginnend mit Zeile 3 aus File 16 und arbeitet es ab Zeile Ø ab.

Load File-Befehl

Mit dem Load File-Befehl (ldf) werden Daten und Programm-Files in den Speicher des Rechners eingeladen.

> HINWEIS
>
> Mit dem ldf-Befehl können, je nach File-Typ, Programme oder Variable eingeladen werden, mit den ldp-und rcf-Befehlen können, je nach Befehl in bestimmte Speicherbereiche eingeladen oder Bereiche aufgezeichnet werden. Mit dem ldf-Befehl können dadurch fälschlicherweise Programme eingeladen werden, wenn Variable übernommen werden sollten oder umgekehrt.

Einladen von Programmen

Syntax:

`ldf` [File-Nummer [,1. Zeilennummer [,2. Zeilennummer]]]

Mit dem Load File-Befehl (ldf) zusammen mit diesen Angaben werden Programme aus der angegebenen File in den Speicher des Rechners eingeladen. Dieser Befehl ist dem ldp-Befehl ähnlich, außer daß mit ldf ein Programm weiterläuft, und mit ldp ein Programm gestartet wird.

Eingabe über die Tastatur

Dieser Befehl wird über die Tastatur wie folgt ausgeführt. Sind keine Parameter angegeben, so wird das Programm von File Ø beginnend mit Zeile Ø eingeladen. Wird eine File-Nummer angegeben, so wird diese File beginnend mit Zeile Ø eingeladen. Sind die File-Nummer und eine Zeilennummer angegeben, so wird diese File beginnend mit der angegebenen Zeilennummer eingeladen. Sind alle drei Parameter angegeben, so wird die angegebene File beginnend mit der ersten Zeilennummer eingelesen und das Programm läuft automatisch mit der zweiten Zeilennummer weiter (alle Variablen bleiben erhalten; „ldp" löscht alle alten Variablen; s. Continue-Befehl).

In einem Programm

Dieser Befehl wird in einem Programm wie folgt ausgeführt. Sind keine Parameter angegeben, so wird das Programm von File Ø beginnend mit Zeile Ø eingeladen und das Programm läuft automatisch ab Zeile Ø weiter.

Ist die File-Nummer angegeben, dann wird das Programm aus dieser File beginnend mit Zeile Ø eingeladen und ab Zeile Ø abgearbeitet. Sind die File-Nummer und eine Zeilennummer angegeben, so wird die angegebene File beginnend mit der angegebenen Zeile eingeladen und das Programm läuft ab dieser Zeile.

Sind alle drei Parameter angegeben, so wird der Befehl genauso wie bei der Eingabe über die Tastatur ausgeführt. Das bedeutet, daß praktisch mit der zweiten Zeilennummer ein Continue -Befehl ausgeführt wird. Alle drei Parameter können auch Ausdrücke sein.

Dieser Befehl darf nicht nach einem Enter-Befehl verwendet werden oder in der Betriebsart „Live Keyboard", um eine Programm-File einzulesen.

Beispiel:

```
ldf 1
13: ldf 2
```

File 1 beginnend mit Zeile Ø einladen.
File 2 beginnend mit Zeile Ø einladen
und ab Zeile Ø abarbeiten (wie Continue).

Verbinden von Programmen

Sind Programme zu lang, um im Rechner gespeichert zu werden, so können sie aufgeteilt und in mehreren Files gespeichert werden. Die Programmteile werden durch das Programm mit dem ldf-Befehl nacheinander eingelesen. Variable, Flags und Unterprogramm-Rücksprungadressen bleiben dabei erhalten.

Im folgenden Beispiel werden 3 Segmente verwendet. Jedes Segment wird dann eingeladen, wenn es erforderlich ist.

Die Segmente rufen sich gegenseitig auf.

Programmsegment in File Ø

```
0: prt "file 0";
 π→A
1: ldf 1
```

Programmsegment in File 1

```
0: prt "file 1";
 prt A
1: ldf 2
```

Programmsegment in File 2

```
0: prt "file 2"
1: end
```

Drücken: LOAD 0 ( ) RUN

```
file 0
file 1
              3.1416
file 2
```

Einlesen von Daten

Syntax:

`ldf` [File-Nummer [, Daten-Liste]]

Mit dem „Load-File"-Befehl (ldf) werden Daten aus der angegebenen File der eingestellten Spur in den Rechner eingelesen. Die Datenliste enthält die durch Komma getrennten Variablen. Einfache und Feld-Variable dürfen mit r-Variablen nicht im gleichen ldf-Befehl verwendet werden.

Ist keine Variablen-Liste angegeben, so werden alle Daten als r-Variable von rØ anfangend übernommen, bis alle Daten eingelesen sind. Ist eine r-Variable angegeben, werden alle Daten beginnend mit dieser r-Variablen übernommen, bis alle Daten eingelesen sind. Werden zwei r-Variable angegeben, so werden die Daten ab der ersten r-Variablen bis zur zweiten angegebenen Variablen übernommen. Stehen mehr Daten zur Verfügung als r-Variable angegeben sind, werden keine Daten übernommen.

Werden einfache oder Feld-Variable angegeben, so werden die Daten beginnend mit der ersten Variablen übernommen bis allen Variablen Werte zugeordnet sind. Sind mehr Werte als Variable vorhanden, so werden keine Werte eingelesen. Sind weniger Werte vorhanden als Variable, so werden alle Daten eingelesen. Variable müssen hintereinander stehen.

Beispiel:

```
ldf  4,r2,r10     Daten von File 4 von r2 bis r10 einlesen.
ldf  r12,A,B[*]   Die durch r12 bezeichnete Daten-File einlesen und
                  der Variablen A und dem Feld B zuordnen.
```

Speicherung von Feld- und r-Variablen

r-Variable werden in der umgekehrten Reihenfolge der Feld-Variablen gespeichert. Werden deshalb r-Variable gespeichert und dann zurück in ein Feld geladen, so ist ihre Reihenfolge umgekehrt.

Beispiel:

```
0: 1→r1;2→r2;
 3→r3;4→r4;5→r5
1: rcf 0,r5
2: dim A[6]
3: ldf 0,A[*]
4: 1→I
5: prt A[I];jmp
 (I+1→I)>6
6: end
```

```
r5— A [1]    5.0000
r4— A [2]    4.0000
r3— A [3]    3.0000
r2— A [4]    2.0000
r1— A [5]    1.0000
r0— A [6]    0.0000
```

In Zeile 1 werden die r-Variablen 0 bis 5 in File Ø gespeichert. In der Zeile 3 wird das Feld A von File Ø geladen. A(5) wird zuerst eingeladen; A(1) wird zuletzt eingeladen.

Record Key-Befehl

Syntax:

```
rck     [File-Nummer]
```

Mit dem Record Key-Befehl (rck) werden die Funktionen der frei definierbaren Funktionstasten in der angegebenen File auf der eingeschalteten Spur gespeichert. Wird keine File-Nummer angegeben, so wird in File Ø gespeichert. Die File-Nummer kann auch ein Ausdruck

sein. Die angegebene File muß vorher markiert werden, ehe der Befehl „Record Key" ausgeführt werden kann.

Beispiel:

```
rck2
```

Die Funktionen der frei definierbaren Funktionstasten in File 2 speichern.

```
3: rck A[12]
```

Die Funktionen der frei definierbaren Funktionstasten in der File speichern, die durch das zwölfte Element des Feldes A bestimmt ist.

Load Keys-Befehl

Syntax:

`ldk` [File-Nummer]

Mit dem Befehl Load Keys (ldk) werden die Funktionen der frei definierbaren Funktionstasten aus der bezeichneten File in den Rechner eingelesen. Wird die File-Nummer weggelassen, wird von File Ø eingelesen. Die File-Nummer kann auch ein Ausdruck sein. Wird der Befehl Load Keys über das Tastenfeld eingegeben, so werden die Unterprogramm-Rücksprungadressen gelöscht und der Programmzähler wird auf Zeile Ø gestellt.

Dieser Befehl ist in einem Enter-Befehl oder in der Betriebsart „Live Keyboard" nicht zulässig.

Beispiel:

```
ldk4
```

Die Funktionen der frei definierbaren Funktionstasten von File 4 einlesen.

Record Memory-Befehl

Syntax:

```
rcm      [File-Nummer]
```

Mit dem Record Memory-Befehl (rcm) wird der gesamte Speicherinhalt mit Programmen, Daten, Tastenfunktionen, Rücksprungadressen, usw. in der angegebenen File auf der eingestellten Spur der Magnetbandkassette abgespeichert.

Wird die File-Nummer weggelassen, wird automatisch in File Ø abgespeichert. Die File-Nummer kann auch ein Ausdruck sein.

Ein Sonderfall ist, wenn der Speicher ein Binärprogramm enthält. In den mit jedem binären Programm gelieferten Informationen, wird die Bedienung des Rechners zur Ausführung des Record Memory-Befehls erklärt.

Load Memory-Befehl

Syntax:

```
ldm      [File-Nummer]
```

Mit dem Load Memory-Befehl (ldm) wird der vorher aufgezeichnete Speicherinhalt wieder in den Speicher zurückgelesen. Nach dem Einlesen befindet sich der Rechner im gleichen Zustand wie zum Zeitpunkt der Aufzeichnung. Ohne Angabe einer File-Nummer wird automatisch File Ø eingeladen. Läuft gerade ein Programm, wenn der Record Memory-Befehl (rcm) erfolgt, so läuft dieses Programm mit dem nächsten Befehl weiter, nachdem die Information in den Speicher übernommen ist.

Die Befehle Record Memory und Load Memory eignen sich besonders zur Eingabe über das „Live Keyboard" oder über eine spezielle Funktionstaste, um den momentanen Speicherinhalt das Rechners aufzuzeichnen, ohne das laufende Programm zu verändern. Dieser Befehl wird öfter verwendet, wenn mit häufigen Stromausfällen zu rechnen ist.

Die File-Nummer kann auch ein Ausdruck sein.

Load Binary Program-Befehl

Syntax:

`ldb` [File-Nummer]

Mit dem Befehl „Load Binary Program" (ldb) werden binäre Programme* in den Speicher des Rechners von der angegebenen File von der Bandkassette eingelesen. Solche im Speicher enthaltene binäre Programme können jederzeit mit anderen binären Programmen überschrieben werden.

Wird keine File-Nummer angegeben, wird von File Ø eingelesen. Die File-Nummer kann auch ein Ausdruck sein.

Beispiel: `ldb2` Das binäre Programm von File 2 einlesen.

Da binären Programmen ein spezieller Platz im Speicher zugewiesen ist, müssen beim Einlesen bestimmte Regeln beachtet werden.

Sind keine einfachen oder Feld-Variablen zugeordnet, so können binäre Programme jederzeit eingelesen werden, falls genügend Speicherplatz vorhanden ist.

Sind jedoch schon einfache oder Feld-Variable zugeordnet, so kann ein binäres Programm nicht eingeladen werden, wenn nicht durch ein früher eingelesenes binäres Programm Speicherplatz reserviert ist.

Beim Einladen sollte folgendes beachtet werden: Bevor einfache oder Feld-Variable angegeben werden, wird das größte zur Verfügung stehende binäre Programm eingeladen. Darauf können Variable zugeordnet werden und binäre Programme können ohne Rücksicht auf Speicherplätze anstelle des schon eingelesenen Programms gespeichert werden.

*Ein binäres Programm ist in Maschinensprache geschrieben, es kann nicht ausgedruckt oder angezeigt werden.

Überprüfung der aufgezeichneten Files

Die aufgezeichneten Files können nochmals mit dem Speicherinhalt verglichen werden um Fehler in der Aufzeichnung feststellen zu können, ohne daß der Speicherinhalt verlorengeht. Wird bei der Überprüfung ein Fehler festgestellt (Fehler 44), sollte die File nochmals aufgenommen werden. Wiederholt auftretende Fehler können auf ein beschädigtes Band zurückzuführen sein.

Zur Überprüfung der Aufzeichnung ist ein stärkeres Signal vom Band erforderlich als zum Einlesen in den Rechner. Dadurch wird sichergestellt, daß eine File später zuverlässiger in den Rechner eingelesen werden kann, wenn sie überprüft ist.

Der Rechner wird automatisch auf Überprüfung geschaltet, wenn das Gerät eingeschaltet wird, der Befehl Erase a ausgeführt wird oder die Taste RESET gedrückt wird. Mit zwei Befehlen kann die automatische Überprüfung ein- oder ausgeschaltet werden.

Auto-Verify Disable-Befehl

Syntax: `avd`

Mit dem Befehl „Auto-Verify Disable" (avd) wird die automatische Überprüfung der aufgezeichneten Files abgeschaltet.

Beispiel:

```
12: avd
```

Schaltet die automatische Überprüfung der aufgezeichneten Files ab.

Auto-Verify Enable-Befehl

Syntax: `ave`

Mit dem Befehl „Auto-Verify Enable" (ave) wird die automatische Überprüfung der Files eingeschaltet. Nachdem dieser Befehl eingegeben ist, werden alle auf Band gespeicherten Informationen automatisch überprüft. Diese Betriebsart der automatischen Überprüfung ist auch eingeschaltet, nachdem der Rechner eingeschaltet ist, RESET gedrückt wird oder der Befehl Erase a ausgeführt wird.

Beispiel: `ave` Schaltet die automatische Überprüfung ein.

Verify-Befehl

Syntax:

```
vfy      [Überprüfungsvariable]
```

Mit dem Befehl „Verify" (vfy) werden die auf Band gespeicherten Informationen mit dem Speicherinhalt des Rechners verglichen. Ist der Inhalt des Speichers mit dem Fileinhalt identisch, so ist der Wert der Überprüfungsvariablen Ø. Werden Unterschiede festgestellt, ist die Überprüfungsvariable 1. Sind die beiden Informationen nicht identisch und ist keine Überprüfungsvariable angegeben, wird Fehler 44 angezeigt.

Die Überprüfungsvariable kann eine einfache, eine Feld- oder einer r-Variable sein.

Der Verify-Befehl kann jedem Aufzeichnungsbefehl folgen, außer Record Memory. Folgt ein Verify (vfy) auf einen (rcm) Befehl, wird Fehler 44 angezeigt. Speicherfiles können nur automatisch überprüft werden.

Dieser Befehl sollte immer verwendet werden, wenn alte Bänder verwendet werden. In diesem Fall wird anstelle der automatischen Überprüfung der Verify-Befehl verwendet. Wird ein Fehler festgestellt, so sollte die Registrierung ein zweites Mal erfolgen.

Set Select Code-Befehl

Der Select Code-Befehl (ssc) ist zum Aufruf späterer Peripherigeräte vorgesehen und sollte nicht verwendet werden. Der Select Code 1 gilt für die interne Kassette.

ANHANG A
SYNTAX

In diesem Anhang wird der Syntax für Modell 9825A beschrieben:

Syntax	Beschreibung	Seite
abs Ausdruck	Absolutwert	63
acs Ausdruck	arc cos	66
[Ausdruck] + [Ausdruck]	Addition	60
[Ausdruck] and [Ausdruck]	AND Operator	62
asn Ausdruck	arc sin	66
Ausdruck → Variable	Zuordnung	59
atn Ausdruck	arc tan	66
avd	Befehl zur Abschaltung der automatischen Überprüfung	126
ave	Automatische Überprüfung	126
beep	Akustisches Signal	55
cfg [Flag Nummer , ...]	Clear Flag	70
cmf [Flag Nummer , ...]	Flag-Funktion umkehren	71
cont [Zeilennummer oder label]	Continue-Befehl	92
cos Ausdruck	cos	66
csv	Einfache Variable löschen	82
deg	Altgrad	65
del erste Zeilennummer [, letzte Zeilennummer] [,*]	Löschbefehl	92

Syntax	Beschreibung	Seite
`dim` Angabe [, Angabe, ...] Angabe kann sein: einfache Variable Feld-Variable [Dimension [, Dimension, ...]]	Dimensionierungsbefehl	80
Ausdruck / Ausdruck	Division	60
`drnd` (Ausdruck, Anzahl der Stellen)	Rundung	63
`dsp` [Kombination von Text und Ausdrücken]	Anzeigebefehl	49
`end`	End-Befehl	57
`enp` [Bezeichnung,] Variable [, [Bezeichnung,] Variable . . .]	Enter print Befehl	54
`ent` [Bezeichnung,] Variable [,[Bezeichnung,] Variable ...]	Enter Befehl	52
Ausdruck = Ausdruck	Vergleichsoperator	61
`erase` [a oder v oder k oder spezielle Funktionstaste]	Löschbefehl	94
`ert` Filenummer	Befehl zum Löschen von Bändern	114
Ausdruck `xor` Ausdruck	Exklusives ODER	62
`exp` Ausdruck	Exponentialfunktion	65
Ausdruck ↑ Ausdruck	Exponent	60
`fdf` [Filenummer]	Suchen einer File	106
`fetch` [Zeilennummer oder spezielle Funktionstaste]	Fetch-Befehl	94
`fla` Flag-Nummer	Flag-Funktion	71
`flt` [Anzahl der Nachkommastellen]	Gleitkomma	48
`frc` Ausdruck	Gebrochener Anteil	63
`fxd` [Anzahl der Nachkommastellen]	Festkomma	46
`grad`	Neugrad	65

Syntax	Beschreibung	Seite
Ausdruck > Ausdruck	Größer als	61
Ausdruck>=Ausdruck	Größer als oder gleich	61
Ausdruck=>Ausdruck	Gleich oder größer als	61
gsb Zeilennummer	Absoluter go sub-Befehl	77
gsb Marke	go sub-Befehl zu einer Marke	79
gsb+ Anzahl der Zeilen	Relativer go sub-Befehl	77
gsb- Anzahl der Zeilen	Relativer go sub-Befehl	77
gto Zeilennummer	Absoluter go to-Befehl	74
gto Marke	go to-Befehl zu Marke	75
gto+ Anzahl der Zeilen	Relativer go to-Befehl	74
gto- Anzahl der Zeilen	Relativer go to-Befehl	74
idf [File-Nummer [, File-Typ [, tatsächliche File-Größe [, absolute File-Größe [, Spur-Files-Nummer]]]]]	Befehl zum Identifizieren von Files	105
if Ausdruck	if-Befehl	79
int Ausdruck	Integer Funktion	63
jmp Anzahl der Zeilen	Sprungbefehl	75
ldb [File-Nummer]	Binäres Programm einladen	125
ldf [File Nummer [, Datenliste]]	Einlesebefehl für Daten-File	121
ldf [File Nummer [, Zeilennummer$_1$ [, Zeilennummer$_2$]]]	Einlesebefehl für Programm-File	119
ldk [File-Nummer]	Einlesen von Tastenfunktionen	123
ldm [File-Nummer]	Speicherinhalt auf Band aufzeichnen	124
ldp [File-Nummer [, Zeilennummer$_1$ [, Zeilennummer$_2$]]]	Programm auf Band aufzeichnen	118

Syntax	Beschreibung	Seite
Ausdruck < Ausdruck	Kleiner als	61
Ausdruck <= Ausdruck	Kleiner gleich	61
Ausdruck =< Ausdruck	Gleich kleiner	61
list k	Funktionen der frei definierbaren Tasten ausdrucken	82
list spezielle Funktionstaste	Funktion einer frei definierbaren Taste ausdrucken	82
list [erste Zeilennummer [, letzte Zeilennummer]]	Ausdruck-Befehl	82
lkd	„Live Keyboard" abschalten	100
lke	„Live Keyboard" einschalten	99
ln Ausdruck	ln	65
log Ausdruck	log	65
max (Liste der Ausdrücke und Felder)	Maximumfunktion	64
min (Liste der Ausdrücke und Felder)	Minimumfunktion	63
Ausdruck mod Ausdruck	Modulus	60
mrk Anzahl der Files, File-Größe [, return Variable]	Markierungsbefehl (Band)	108
Ausdruck * Ausdruck	Multiplikation	60
[Ausdruck] [Ausdruck]	Implizierte Multiplikation	
nor [erste Zeilennummer [, letzte Zeilennummer]]	Normal-Befehl	89
Ausdruck # Ausdruck	Ungleich	61
Ausdruck <> Ausdruck	Ungleich	61
Ausdruck >< Ausdruck	Ungleich	61
not Ausdruck	NOT Operator	62
Ausdruck or Ausdruck	OR Operator	62

Syntax	Beschreibung	Seite
prnd (Ausdruck , Rundungsangabe)	Rundung (auf Potenz von 10)	63
prt [Kombination von Text und Ausdrücken]	Druck-Befehl	50
rad	Bogenmaß	65
rcf File-Nummer, Datenliste	In angegebener File aufzeichnen (Daten)	116
rcf [File-Nummer [, erste File-Nummer [, letzte File-Nummer]] [, "SE" or "DB"]]	In angegebener File aufzeichnen (Programme)	115
rck [File-Nummer]	Funktionen der speziellen Funktionstasten aufzeichnen	122
rcm [File-Nummer]	Speicherinhalt aufzeichnen	124
ret	return (Rücksprung) Befehl	77
rew	Band zurückspulen	104
rnd [–] Ausdruck	Zufallszahlen	64
run [Zeilennummer oder Marke]	Run Befehl	91
sfg [Flag-Nummer , ...]	Flag setzen	70
sgn Ausdruck	Vorzeichen	63
sin Ausdruck	sin	66
spc [Anzahl Leerzeilen]	Zeilenabstand	55
√ Ausdruck	Quadratwurzel	63
ssc Select Code	Select Code	127
stp	Stop Befehl	57
stp erste Zeilennummer [, letzte Zeilennummer]	Stop Befehl (Programmkorrektur)	89
[Ausdruck] – Ausdruck	Subtraktion	60
tan Ausdruck	tan	66
tlist	Druckt Inhaltsverzeichnis für Band aus	107

Syntax	Beschreibung	Seite
`tnt`	Exponentialfunktion 10 ↑	65
`trc` [erste Zeilennummer [, letzte Zeilennummer]]	Trace-Befehl	88
`trk` Spur-Nummer	Spur-Befehl	105
`units`	Ermittelt Winkelmaß	65
`vfy` [Return Variable]	Verify-Befehl	127
`wait` Anzahl der Millisekunden	Wait-Befehl	56

ANHANG B
FEHLERHINWEISE

Tritt in einem Programm ein Fehler auf, wird der Programmzähler wieder auf Zeile Ø gesetzt. Durch Drücken von [CONTINUE] wird das Programm wieder ab Zeile Ø abgearbeitet. Durch Drücken von cont mit einer Zeilennummer läuft das Programm ab der angegebenen Zeile weiter.

Fehlerhinweise für das Grundgerät

`error 00` Systemfehler. Die Operation, die den Fehler hervorgerufen hat, wird wiederholt. Tritt der Fehler wieder auf, setzen Sie sich bitte mit unserem nächsten HP-Verkaufsbüro in Verbindung.

`error 01` Unerwartete Unterbrechung durch ein Peripherie-Gerät. Dieser Fehler tritt nur auf, wenn ein Peripherie-Gerät angeschlossen ist. Zum Löschen des Fehlers Taste [RESET] drücken.

`error 02`* Text ist nicht abgeschlossen. Die Textzeile muß mit einem Anführungszeichen abgeschlossen werden.

`error 03`* Abkürzung ist nicht bekannt. Dieser Fehler tritt meist bei einem Tippfehler auf, wenn z. B. go to an Stelle von gto geschrieben wird oder wenn eine Anweisung in Betriebsart „Live Keyboard" oder in einem Enter-Befehl eingegeben wird.

`error 04` Geschütztes Programm. Fehlerhinweis erfolgt, wenn Zeilen eines geschützten Programms angezeigt oder ausgedruckt werden sollen.

`error 05` Unerlaubte Operation. Die Zeile kann nicht mit einer Zeilennummer gespeichert oder ausgeführt werden. Dieser Fehler kann z. B. auftreten, wenn eine Zeile zur Korrektur in die Anzeige geholt ist und die Tasten [EXECUTE], [STORE] oder Line [INSERT] gedrückt werden.

* Bei diesen Fehlern wird der Fehler selbst durch eine blinkende Marke angezeigt.

error 06 *Zeile enthält einen Syntax-Fehler.

error 07 *Eingegebene Zeile enthält Syntax-Fehler. Z. B.: gtoprt5

error 08 Die Zeile ist zu lang
(manchmal erscheint die blinkende Marke)

error 09 Die Befehle gto, gsb oder end sind hier nicht zulässig, z.B. die Ausführung eines End-Befehls auf einen ent-Befehl.

error 10 *Bei go to oder go sub-Befehlen sind Ganzzahlen erforderlich. zum Beispiel
gto23.4 ist nicht zulässig.

error 11 Ganzzahl liegt außerhalb des Bereichs oder Ganzzahl ist erforderlich, muß zwischen –32768 und +32767 liegen.

Beispiel:	spc50000	Ganzzahl liegt außerhalb des Bereichs.
	del A	Ganzzahl ist erforderlich.

error 12 *Diese Zeile kann nicht gespeichert werden. Sie läßt sich nur ausführen.

Beispiel: 2+2 (R/S) Wird zwar berechnet, ist jedoch nicht zulässig.
2+2 (STORE)

error 13 Enter-Befehl ist hier nicht zulässig. Zum Beispiel darf entX nicht über die Tastatur eingegeben werden; wird nur im Programm verwendet.

error 14 Das Programm ist gestört. Dieser Fehler kann auftreten, wenn die (RESET) Taste gedrückt wird, wenn ein Programm geändert werden soll. Es sollten dann die Daten aufgezeichnet und der Befehl Erase a ausgeführt werden.

error 15 Druckerpapier zu Ende oder Störung des Druckers.

error 16 Zur Zeichenkettenverarbeitung, z.B. das Zeichenketten-ROM ist nicht eingesetzt oder das Argument bei einem logischen Vergleich ist nicht zulässig. Ist das Zeichenketten-ROM nicht eingesteckt und erfolgt folgende Eingabe:

if "B"<"A" wird Fehler 16 angezeigt.

* Bei diesen Fehlern wird der Fehler selbst durch eine blinkende Marke angezeigt.

error 17 Parameter liegt außerhalb des Bereichs.

Beispiel:
```
wait-5
fxd15
```

error 18 Parameter ist nicht zulässig.

Beispiel:
```
erase z
```

error 19 Zeilennummer ist falsch.

Beispiel:
```
del 10,5
```

error 20 Ein ROM ist nicht eingesteckt. Dadurch läßt sich die Eingabe nicht ausführen. Dieser Fehler tritt meistens auf, wenn [↑] oder [↑] gedrückt oder List eingegeben wird. Die rechte angezeigte Zahl gibt das fehlende ROM an. Beim Programmablauf wird nur die Zeilennummer angezeigt.

Angezeigte Nr. des ROM	ROM
1	Binärprogramm
6	Zeichenketten-ROM
8	Erweitertes I/O-ROM
9	Erweiterte Programmierung-ROM
10	Matrix-ROM
11	Plotter 9862A ROM
12	Universal I/O-ROM

error 21 Die Zeile ist zu lang, um abgespeichert zu werden. Dies kann auftreten, wenn Leertasten oder Klammern automatisch vom Rechner eingefügt werden. Zum Beispiel werden automatisch Klammern bei der Speicherung der folgenden Zeile eingefügt: `tan2→A`, die wie folgt abgespeichert wird `tan(2)→A`

error 22 Die angegebene Dimensionierung ist nicht zulässig. Dieser Fehler wird z. B. angezeigt, wenn die untere Grenze eines Feldes größer ist als die obere Grenze. Dieser Fehler wird auch angezeigt, wenn eine Zeichenkette dimensioniert wird, obwohl das Zeichenketten-ROM nicht in den Rechner eingesteckt ist.

error 23 Die einfache Variable ist bereits zugeordnet.

Beispiel: `2→X;dimA[5],X`

error 24 Das Feld ist bereits dimensioniert.

Beispiel: `dimA[4],B[5],A[6]`

error 25 Die Dimensionierung des Feldes stimmt mit den Elementen nicht überein.

Beispiel: `dimX[2,7];1→X[5]`

error 26 Die Elemente eines Feldes liegen außerhalb des Bereichs.

Beispiel: `dimA[12];2→A[58]`

error 27 Feld ist nicht definiert. Das Feld muß zuerst in einem Dimensionierungsbefehl enthalten sein.

error 28 Dem Return-Befehl entspricht kein gsb-Befehl.

error 29 Die Zeile wird nicht ausgeführt, da ein ROM fehlt.

Beispiel: plt-Befehl, ohne daß das Plotter-Rom eingesteckt ist.

error 30 Die Funktion der frei definierbaren Funktionstaste ist nicht definiert.

error 31 Diese Programmzeile existiert nicht.

Beispiel: gto 900-Befehl in einem Programm mit 5 Zeilen.

error 32 Falscher Datentyp, es ist eine Zahl erforderlich.

error 33 Diese Daten stimmen nicht mit dem Zuordnungsbefehl überein.

error 34 Überlauf der Anzeige, da eine spezielle Funktionstaste gedrückt wurde, deren Funktion für die Zeile zu lang ist.
Eine Zeile darf nur bis 80 Zeichen enthalten.

error 35 Bezug auf die Flag ist falsch. Diese Flag existiert nicht.

Beispiel: `sf 18`

error 36 Versuch, die Adresse eines gto oder gsb-Befehls zu löschen. Die Operation wird nicht ausgeführt.

error 37 Überlauf des Pufferspeichers für die Anzeige; durch den Display-Befehl verursacht.

error 38 Nicht genügend Speicherplatz für die Unterprogramm-Rücksprungadresse.

error 39 Nicht genügend Speicherplatz zur Zuordnung von Variablen oder für binäre Programme. Die Zuordnung wird nicht ausgeführt.

error 40 Nicht genügend Speicherplatz zur Ausführung der Operation.

Beispiel: wenn eine Zeile gespeichert werden soll, für die der Speicher nicht ausreicht.

error 41 Magnetbandkassette ist nicht eingesetzt.

error 42 Die Magnetbandkassette ist gegen Überschreiben geschützt. Der Schieber an der Kassette muß zur Aufzeichnung nach rechts geschoben werden.

error 43 Markierung des Bandanfangs oder des Bandendes oder Band wird nicht richtig transportiert.

error 44 Bei der Überprüfung der Registrierung wurde Fehler festgestellt (s. S. 126).

error 45 Bei unbekannter Stellung des Magnetbandes soll ein idf-Befehl ohne Parameter oder ein mrk-Befehl ausgeführt werden.

error 46 Lesefehler, wenn eine File eingelesen werden soll. Der Teil mit dem Fehler wird nicht eingelesen (s. Anhang F).

error 47 Lesefehler des File-Kopfes (s. Anhang F)

error 48 Das Bandende ist erreicht, ehe die angegebene Anzahl der Files markiert wurde.

error 49 File ist zu klein.

error 50 Der ldf-Befehl für eine Programm-File muß der letzte Befehl einer Zeile sein.

error 51 Es ist ein ROM eingesteckt, das während des Record Memory-Befehls nicht eingesteckt war. Das durch die folgende, rechts neben dem Fehlerhinweis erscheinende Ziffer angegebene ROM ist herauszuziehen und der Load Memory-Befehl nochmals auszuführen.

Ziffer	ROM
1	Binärprogramm
6	Zeichenketten-ROM
8	Extended I/O-ROM
9	Erweiterte Programmierung-ROM
10	Matrix-ROM
11	Plotter-ROM
12	Universal I/O-ROM

error 52 Das in der vorigen Tabelle mit einer Ziffer bezeichnete ROM war eingesetzt, als der Record Memory-Befehl ausgeführt wurde, ist aber jetzt nicht eingesteckt. Das ROM ist wieder einzusetzen und der Load Memory-Befehl nochmals einzugeben.

error 53 Negativer Parameter bei Kassetten-Befehl.

Beispiel:

```
mrk-12.300
trk-1
```

error 54 Das einzulesende Binärprogramm ist größer als das augenblicklich geladene Binärprogramm und Variable sind zugeordnet.

error 55 Falsche oder keine Parameter in einem Befehl für die Bandkassette.

error 56 Für den Befehl für die Bandkassette ist die Liste der Daten nicht hintereinander im Speicher enthalten.

error 57 Falscher File-Typ. Dieser Fehler kann z. B. auftreten, wenn versucht wird, ein Programm von einer Daten-File oder Tasten-File einzuladen.

error 58 Unzulässiger Parameter in einem rcf-Befehl; es sind nur "SE" oder "DB" zulässig.

error 59 Es wird versucht, ein nicht existierendes Programm, nicht existierende Daten oder Funktionen der frei definierbaren Funktionstasten aufzuzeichnen.

error 60 Es wird versucht, eine leere File oder eine Ø-File (Typ Ø) einzulesen.

error 61 Die in einem ldf-oder ldp-Befehl angegebene Zeile existiert nicht.
Wurde die Zeile mit dem ldf- oder ldp-Befehl durch den Load-Befehl überschrieben, kann die angezeigte Zeilennummer falsch sein.

error 62 Der angegebene Speicherplatz ist für die einzulesende File der Kassette zu klein.

error 63 Durch das Einlesen von der Kassette würde die gsb-Unterprogramm-Rücksprungadresse im Programm überschrieben. Es wird nicht von der Kassette eingelesen.

error 64 Bei der Betriebsart „Live Keyboard" oder in einem Enter-Befehl werden die Befehle ldk, ldf oder ldp eingegeben.

error 65 Die Filenummer wird nicht gefunden. Die angegebene Filenummer in fdf-Befehl existiert nicht.

Bei den Fehlern 66 bis 77 wird ein Wert zugewiesen und kein Fehler angezeigt, wenn Flag 14 gesetzt ist.

error 66 Division durch Ø. Zugewiesener Wert + oder – 9.99999999999 e 511. A mod B, wobei B gleich Ø ist. Zugewiesener Wert = Ø.

error 67 Quadratwurzel einer negativen Zahl.
Zugewiesener Wert: $\sqrt{(\text{abs (Argument)})}$

error 68 Tan (n$*\pi$/2 Bogenmaß)
Tan (n$*$90 Altgrad)
Tan (n$*$100 Neugrad)
wobei n ein ungerade ganze Zahl ist.

Zugewiesener Wert = +9.99999999999e 511, für n > Ø.
Zugewiesener Wert = –9.99999999999e 511, für n < Ø.

error 69 In oder log einer negativen Zahl. Zugewiesener Wert = In (abs (Argument)) oder log (abs (Argument))

error 70 In oder log von Ø. Zugewiesener Wert = 9.99999999999e 511.

error 71 asn oder acs einer Zahl, kleiner als −1 oder größer als +1. Zugewiesener Wert: asn (sgn (Argument)) oder acs (sga (Argument))

error 72 Negative Basis mit gebrochenem Exponenten. Zugewiesener Wert: (abs (Basis)) ↑ (gebrochener Exponent)

error 73 Ø zur Potenz von Ø (Ø ↑ Ø). Zugewiesener Wert: 1

error 74 Überlauf. Zugewiesener Wert: + oder −9.99999999999e99.

error 75 Zahl ist zu klein. Zugewiesener Wert: Ø.

error 76 Überlauf eines Zwischenergebnisses. Zugewiesener Wert: + oder −9.99999999999e 511.

error 77 Zwischenergebnis zu klein. Zugewiesener Wert: Ø.

Fehlermeldungen des ROMs für erweiterte Programmierung

error A0 Operatoren wie <, > sind bei einem for-Befehl nicht erlaubt oder kein Apostroph nach Unterprogramm-Bezeichnung.

error A1 Ein for-Befehl wird nicht durch einen next-Befehl abgeschlossen.

error A2 Es wird ein next-Befehl durchlaufen, für den der for-Befehl fehlt.

error A3 Ein nicht numerischer Parameter wird als p-Wert übergeben.

error A4 Kein return-Parameter für ein Funktions-Unterprogramm.

error A5 Falsche p-Wert-Referenz, es werden keine Funktionen oder Unterprogramme ausgeführt.

error A6 Versuch, lokalen p-Wert über das Tastenfeld zuzuordnen.

error A7 Falsche Anzahl von Parametern in fts-, stf-, fti-oder itf-Funktionen. Parameter für stf oder itf müssen Zeichenketten sein (nicht numerisch). Die Parameter für stf oder itf enthalten zu wenig Zeichen.

error A8 Überlauf oder Unterlauf bei fts-Funktion oder Überlauf bei fti-Funktion.

error A9 Zeichenketten ROM ist für stf-oder itf-Funktion nicht eingesetzt.

Diese Fehler des Grundgeräts haben bei eingesetztem ROM für erweiterte Programmierung zusätzliche Bedeutung:

error 09 Versuch, einen next-Befehl über die Tastatur auszuführen, während eine for next-Schleife mit derselben Variablen im Programm ausgeführt wird oder Versuch, eine Funktion oder ein Unterprogramm über die Tastatur aufzurufen.

error 26 p-Wert-Referenz ist negativ.

error 32 Nicht numerischer Wert in einem for Befehl oder nicht numerischer Parameter bei fts-oder fti-Funktion.

error 38 Speicherüberlauf bei Aufruf einer Funktion oder eines Unterprogramms.

error 40 Speicherüberlauf bei einem for-Befehl oder bei der Zuordnung lokaler p-Werte.

Fehlermeldungen des Universal I/O-ROMs

error G1 Falsche Formatangabe:

- Formatangabe im Formatbefehl außerhalb des Bereichs $0 \leq n \leq 9$
- Angegebene Formatnummer existiert nicht.

error G2 Der angegebene Formatbefehl ist fehlerhaft:

- Falsche Formatangabe
- Numerischer Überlauf bei Formatbefehl

error G3 Falsche I/O Parameter:

- Parameter keine Zahl oder Zeichenkette
- Negativer Parameter bei numerischer f z Angabe
- Numerischer Parameter bei c edit Angabe
- Binärparameter nicht im Bereich von $-32768 \leqslant n \leqslant 32767$
- Mehr als ein Parameter in einer read binary-oder read status-Funktion.
- Nicht numerischer Parameter fehlt in einem write control-Befehl.

error G4 Falscher Select-Code:

- Select-Code ist nicht numerisch oder größer als 4 Stellen.
- Select-Code ist beim read status größer als 2 Stellen.
- Select-Code ist nicht im Bereich Ø bis 16.
- Select-Code 1 ist nur beim Read Status erlaubt.
- HP-IB-Adresse nicht im Bereich von Ø bis 31.
- Einlesen bei Select-Code Ø nicht erlaubt.

error G5 Falscher Leseparameter:

- Leseanweisung enthält Konstante.
- Leseoperation füllt die Zeichenkette nicht.
- Numerischer Parameter bezieht sich auf ein c Format.

error G6 Falscher Parameter im Code-Umwandlungsbefehl:

- Mehr als 20 Parameter.
- Ungerade Anzahl von Parametern.
- Nicht numerischer Parameter.
- Parameter nicht im Bereich $\emptyset \leqslant n \leqslant 127$.

error G7 Eingangsdaten werden nicht angenommen:

- Mehr als ein Dezimalpunkt oder „E" wird eingelesen.
- 511 Zeichen werden ohne LF (line feed) eingelesen.
- „E" ohne vorangehende Zahl.
- Es werden mehr als 158 numerische Zeichen eingelesen.

error G8 Peripheriegerät ausgefallen:

- Falsche Status-bits.
- Durch Taste (STOP) Ausführung beendet.

error G9 Fehler im Interface:

- Fehlerhafte HP-IB-Funktion.
- Nicht belegter I/O-Kanal.
- Select-Code stimmt nicht mit Interfacekarte überein
  (Beispiel: wrt 711 wenn Karte 98032A auf 7 steht, oder wrt 6 wenn Karte 98034A auf 6 gesetzt ist).
- Write control wird an eine 98034A HP-IB-Karte adressiert.

Fehlermeldungen des Matrix-ROM

error M1* Syntax-Fehler.

error M2 Falsche Dimensionen. Feld-Dimensionen sind nicht aufeinander oder auf die angegebene Funktion abgestimmt.

error M3 Falsche Redimensionierungsanweisung: die neue Anzahl der Dimensionen darf nicht größer sein als die alte Anzahl.

error M4* Operation ist nicht zulässig. Ein Feld links von → darf nicht auch rechts erscheinen.

error M5 Matrix kann nicht invertiert werden. Die errechnete Determinante = Ø.

* Bei diesen Fehlern wird der Fehler durch eine Leuchtmarke markiert, wenn die Taste RECALL gedrückt wird.

Fehlermeldungen des Plotter 9862A ROM

error P1 Befehle sind in der falschen Reihenfolge angegeben (siehe Anhang des ROM Handbuchs).

error P2 Falsche Anzahl von Parametern.

error P3 Falsche Art von Parametern. Parameter für einen Label-Befehl müssen Ausdrücke, Text oder Zeichenkettenvariable sein (bei Zeichenketten ist das Zeichenketten-ROM erforderlich).

error P4 Skalierung außerhalb des Bereichs. Der Maximalwert ist gleich oder kleiner als der Minimalwert.

error P5 Integer außerhalb des Bereichs. Der Parameter zur Steuerung der Schreibfeder liegt außerhalb des Bereichs von −32768 bis +32767 oder der Select-Code ist nicht Ø und auch nicht 2 bis 15.

error P6 Zeichengröße außerhalb des Bereichs. Die Angabe für die Breite und Höhe der Zeichen im Letter-Befehl ist Ø oder bei der Berechnung der Größe wird die Integergröße überschritten.

error P7 (entfällt)

error P8 Der Ursprungspunkt der Achsen liegt außerhalb der Skalierung. Die angegebenen X- und Y-Werte liegen außerhalb der Zeichenfläche.

Erfolgt die Fehlermeldung PLT DOWN , sind alle Anschlüsse des Plotters zu prüfen und zu prüfen, daß der Plotter eingeschaltet ist und CHART HOLD aktiviert wurde.

Fehlermeldungen des Zeichenketten-ROM

error S0 Falsche Zeichenketten in der Liste der Variablen eines ldf-Befehls.

error S1 Falsches Argument für eine Zeichenketten-Funktion oder -Variable.

error S2 Mehr Parameter als für die Zeichenketten-Funktion oder -Variable zulässig.

error S3 Es wird versucht, einer nicht zusammenhängenden Zeichenkette zuzuweisen oder auf sie zuzugreifen. Es wird versucht, eine numerische Funktion mit einer Null-Zeichenkette auszuführen.

error S4 Versuch, die val-Funktion mit einer nicht numerischen oder Null-Zeichenkette auszuführen. Der Exponent in einer val-Funktion ist zu groß oder nicht zulässiges Format des Exponenten.

error S5 Ungültige Zuordnung zu einer Zeichenkette.

error S6 Falsche Reihenfolge der Parameter einer Zeichenketten-Variablen oder der Parameter ist Ø, negativ oder überschreitet die dimensionierte Größe.

error S7 Die Zeichenkette ist noch nicht zugeordnet.

error S8 Die Zeichenkette ist schon zugeordnet.

error S9 Die maximale Länge der Zeichenkette wird überschritten.

ANHANG C
PROGRAMMIER-HINWEISE

Ein Programm kann verschieden aufgebaut sein, um eine Aufgabe zu lösen. Die Entscheidung liegt dabei beim Programmierer. Es wird jedoch angestrebt, mit wenig Speicherplatz auszukommen, die Ausführungszeit kurz zu halten und das Programm übersichtlich zu gestalten. Zwischen diesen Anforderungen muß ein Kompromiß gefunden werden.

Dieser Abschnitt enthält Hinweise zur Programmierung, um Speicherplatz oder um Zeit einzusparen. Diese Liste ist nur ein Auszug. Die Besonderheiten sind rechnerspezifisch und deshalb nicht sofort offensichtlich.

Einsparungen bei der Ausführungszeit fallen kaum ins Gewicht, wenn die einzelnen Befehle nicht mehrere tausend mal durchlaufen werden. An Speicherplatz lassen sich meist nur wenige Byte einsparen. Um die Anzahl der Speicherplätze für einen Befehl zu ermitteln, wird `list-1` ausgeführt, nachdem der Befehl gespeichert wurde.

Methode A	Methode B	Weniger Speicherplatz	Schnellere Ausführung
Einfache Variable	r-Variable	A	A
r-Variable	Eindimensionale Feld-Variable	gleich	A
Mehrere Befehle in einer Zeile	Ein Befehl pro Zeile	A	A
gto+5	gto 5	gleich	gleich
gto—5	gto 5	gleich	gleich
gto „5" (label mit einem oder zwei Zeichen)	gto 5	gleich	gleich
gto+5	jmp 5 (Hinweis 1)	B	A
$\sqrt{x}$	x ↑ .5	A	A
xx	x ↑ 2 (Hinweis 2)	A	gleich

Methode A	Methode B	Weniger Speicherplatz	Schnellere Ausführung
Implizierte Multiplikation	Angegebene Multiplikation	gleich	gleich
π	3.14159.....	A	A
if flg 2 = 1	if flg 2	B	B
if flg 2 = 0	if not flg 2	B	B
if A # O	if A	B	B
if (A < B) or (B < C)	if (A < B) + (B < C)	gleich	A
if (A < B) and (B < C)	if (A < B) * (B < C)	gleich	A
J+5 → K; K–3 → L	(J+5 → K) –3 → L	B	B
J+1 → J; if J < 5	if (J+1 → J) < 5	B	B
untere Grenze für Feld-Dimensionen angegeben	zugewiesenen Wert verwenden	B	gleich
Verwendung einer einfachen Variablen als Flag (1 A)	Flag	(Hinweis 3)	B
Verwendung beider Spuren abwechselnd	Verwendung der Spuren nacheinander	gleich	A

Hinweis 1: Für berechnete Verzweigungen kann nur der jump (jmp)-Befehl verwendet werden.

Hinweis 2: X↑Y wird durch wiederholte Multiplikation errechnet, wenn Y eine Integerzahl ist.

Hinweis 3: Wird nur ein Test ausgeführt, wird durch Flag weniger Speicher belegt; werden zwei Tests ausgeführt, ist der Speicherplatz gleich. Bei mehr als zwei Tests wird durch die Methode der einfachen Variablen weniger Speicher belegt.

ANHANG D
RECHNER-STATUS

Die folgende Tabelle zeigt den jeweiligen Status des Rechners nach der Durchführung der angegebenen Operation (siehe auch die einzelnen Abschnitte des Handbuchs).

	Operationen					
	Erase all oder nach Einschalten	Reset	Erase	Run	Continue nach Korrekturen	Continue
Variable	R	X	R	R	X	X
Flags Ø bis 15	R	X	R	R	X	X
Ergebnis	R	X	X	X	X	X
Binärprogramm	R	X	X	X	X	X
Unterprogramm-Rücksprungadresse	R	R	R	R	R	X
Betriebsart Print-all	R	R	X	X	X	X
Verify	R	R	X	X	X	X
„Live Keyboard"	R	R	X	X	X	X
Geschütztes Programm	R	X	R	X	X	X
Kassetten-Select-Code	R	R	X	X	X	X
Kassetten-Spur	R	R	X	X	X	X
Winkelmaß für trig. Funktionen	R	R	X	X	X	X
Einstellung Fest-/Gleitkomma	R	R	X	X	X	X
Anfangszahl für Zufallszahlen	R	R	X	X	X	X
Betriebsart Trace	R	R	X	X	X	X

R = Zustand wie nach Einschalten

X = nicht verändert

ANHANG E
UNTERSCHIEDE ZWISCHEN MODELL 9825 A und 9820 A/9821 A

Jedes Programm für die Rechner 9820A/9821A kann auch nach kleinen Änderungen (wie E, Exponent, und Befehle in Kleinschreibung) für Modell 9825A verwendet werden.

Die folgende Liste enthält die kleinen Unterschiede zwischen den Rechnern. Sie ist in zwei Abschnitte eingeteilt: in Unterschiede bei der Eingabe und in Unterschiede bei der Ausführung von Programmen.

Eingabe von Programmen

- Hinter einem Label einer Zeile steht ein Doppelpunkt, bei 9820A/9821A ein Semikolon.

- Klammern werden verwendet, um anzugeben, welcher Vergleich zuerst ausgeführt wird: `if A=B=C;` wird eingegeben als: `if (A=B)=C;` bei Modell 9825A (ist nicht gleich `if A=B and B=C`

- Durch Speichern des end-Befehles werden bei Modell 9825A keine Zeilen mit höheren Nummern gelöscht.

- `r-A` ist bei 9825A nicht zulässig, es muß `r(-A)` verwendet werden.

- Der enter (ent)-Befehl wird von 9825A auf Syntax überprüft. Der Inhalt eines enter-Befehls muß aus Text oder Variablen bestehen. Ausdrücke wie `ent AA→B` sind bei 9825A nicht zulässig; es muß heißen `ent A; AA→B`

- Ketten von Operatoren wie `---X` sind beim 9825A nicht erlaubt.

- File-Größen in einem Markierungsbefehl werden in Byte anstelle von Registern angegeben. Daher wird `MRK 1,X` des 9821A bei Modell 9825A zu `mrk1,8X`.

- Bei Modell 9825A werden Programme anders verbunden, `GTO X; LDF Y` wird bei Modell 9825A zu `ldf Y, X` .

- `A+B→C+X→Y` für die Modelle 9820A/9821A wird auf dem 9825A als `(A+B→C) +X→Y` eingegeben.

- `E-6` wird bei Modell 9825A als `1e-6` eingegeben.

- Die TBL-Funktionen der 9820A/9821A werden wie folgt ersetzt:

9820A/21A	9825A
Funktion	**ersetzt durch**
TBL0	units
TBL1	deg
TBL2	rad
TBL3	grad
TBL4	entfällt
TBL5	csv
TBL6	cfg

Abarbeiten von Programmen

- Beim 9825A werden Vergleiche auf 12 Stellen ausgeführt. Bei den Modellen 9820A/21A wird auf 10 Stellen gerundet und dann verglichen.
  Für: `if X=Y;` wird bei Modell 9825A verwendet: `if drnd(X, 10)=drnd(Y, 10)`

- Bei der Ermittlung von Integerzahlen werden Gleitkommazahlen vom 9825A gerundet, bei 9820A/21A ohne Rundung abgetrennt.
  Hier einige Beispiele für Modell 9825A:
  1. r (4.9) bedeutet r 5
  2. jmp 2.9 entspricht jmp 3
  3. sfg 5.95 entspricht sfg 6 (gilt auch für cfg, flg und cmf).

- Bei einem gto-oder gsb-Label ist bei 9825A vollständige Übereinstimmung erforderlich und nicht nur für die vier letzten Stellen wie bei 9820A/21A.

- Bei den Modellen 9820A/21A ergibt Ø ↑ Ø = Ø. Bei Modell 9825A ergibt Ø ↑ Ø Fehler 73 (der zugewiesene Wert ist 1).

- Bei Modell 9825A sind Zahlen, Ausdrücke oder Befehle zulässige Eingaben nach Abfragen. Ist jedoch `ent A` die Abfrage und wird `10→Z` eingegeben, so wird Flag 13 gesetzt und A behält den ursprünglichen Wert. Im folgenden Beispiel wird kein Wert eingegeben und Flag 13 wird gesetzt.

  `prt X` Druckbefehl
  `Z→X` Zuordnungsbefehl

  Gültige Eingaben sind:

  `A+7↵` Ausdruck
  `(Z→K)` Eingeschlossene Zuordnung

- Flag 13 wird gelöscht, wenn eine Zahl oder ein Ausdruck nach einem enter-Befehl eingegeben wird.

- Wird bei den Modellen 9820A/21A die run program-Taste gedrückt, ohne einen Wert für `ent R(X+1→X)` einzugeben, so wird der Wert für X nicht inkrementiert und RX wird nicht verändert. Bei Modell 9825A wird der Ausdruck `(X+1→X)` ausgeführt, auch wenn kein Wert eingegeben wird.

- Bei Modell 9825A wird die Integer-Funktion als die größte Ganzzahl definiert, die kleiner oder gleich dem Argument ist. Bei den Rechnern 9820A/21A ist es die größte Ganzzahl, die kleiner oder gleich dem Absolutwert des Arguments ist, wobei das Vorzeichen des Ergebnisses gleich dem des Arguments ist.

- Tritt bei Modell 9825A ein Fehler in einem Befehl auf, so wird die gesamte Zeile nicht ausgeführt. Bei den Modellen 9820A/21A werden die übrigen Befehle der Zeile ausgeführt.

- Implizierte Speicherung nach Z wird durch implizierte Speicherung nach result (res) ersetzt. Z unterscheidet sich nicht mehr von anderen einfachen Variablen. Ein Befehl mit implizierter Speicherung kann nicht mehr gespeichert werden — es muß eine Variable angegeben werden. Ein Programm kann auf den Wert in result (res) zugreifen, der Wert in res kann jedoch nicht durch das Programm geändert werden.

- Wird zu einer Zeile verzweigt, deren Nummer höher ist als die letzte Programmzeile, so wird diese Zeile nicht mehr so behandelt, als wäre sie ein end-Befehl.

- Flags werden durch den end-Befehl bei Modell 9825A nicht gelöscht.

- Durch den stop (stp)-Befehl werden Unterprogramm-Rücksprungadressen nicht gelöscht.

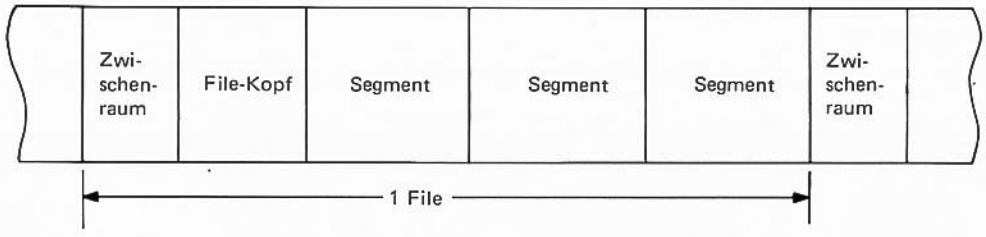
- Bei Modell 9825A wird durch den idf-Befehl das Magnetband immer vor den Filekopf der jeweiligen File positioniert. Durch wiederholte idf-Befehle wird das Band nicht bewegt. Folgt auf den idf-Befehl ein mrk-Befehl, so wird die File neu markiert (alle Informationen in dieser File werden dann gelöscht).

ANHANG F
FEHLERANZEIGE FÜR DIE BANDKASSETTE

File-Einlesefehler

Wird beim Einlesen einer File ein Fehler (46) angezeigt, so werden der Magnetkopf und das Antriebsrad gereinigt, wie auf Seite 103 beschrieben. Darauf wird der Befehl wiederholt, bei dem der Fehler auftrat. Ist der Fehler noch vorhanden, so hängt der nächste Schritt vom File-Typ ab.

Zuerst wollen wir die Struktur der Files auf dem Band betrachten. Eine File setzt sich aus bestimmten Segmenten zusammen. Dadurch ist es möglich, Teile einer File einzulesen, auch wenn ein Fehler angezeigt wird. Fehler 46 zeigt an, daß ein oder mehrere Teile Fehler enthalten können.



Einladen einer Programm-File

Wird der Fehler 46 beim Einladen eines Programmes angezeigt, so können eine oder mehrere Zeilen verlorengehen. Die Stelle, an der der Fehler auftrat, wird im Programm durch ein Sternchen gekennzeichnet. Die fehlenden Zeilen lassen sich dann anhand der Programmliste neu eingeben.

Dabei ist zu beachten, daß die Adressen von go to-und go sub-Befehlen verändert werden. Dadurch kann es möglich sein, die go to und go sub Adressen nach der Eingabe der Zeilen zu korrigieren.

Einladen einer Daten-File

Wird der Fehler 46 beim Einladen numerischer Daten angezeigt, so wird das jeweilige Segment durch „? ? ? ? " gekennzeichnet (bei Float 11-Format). Ein Segment in einer numerischen Daten-File enthält immer 32 Ziffern. Wird eine Eingabe durch „? ? ? ? " ersetzt, können die 31 restlichen Zahlen falsch sein. Diese 31 Zahlen werden wie folgt bestimmt:

- Bei r-Variablen; die 31 höheren r-Variablen können falsch sein.

- Bei einfachen und Feld-Variablen; es ist zu bestimmen, in welcher Reihenfolge die Variablen zugeordnet wurden (siehe Dimensionierungsbefehl). Ab dem durch „? ? ? ? " bezeichneten Element wird dann innerhalb des Dimensionierungsbefehls nach links gegangen. Bei einem Feld hat das erste Element in dem verlorengegangenen Abschnitt die größte Bezeichnung. Durch Verminderung der ganz links stehenden Bezeichnung eines Feldes kann der fehlende Wert gefunden werden. Zum Beispiel in dem folgenden Dimensionierungsbefehl ist ein Segment verlorengegangen:

```
0: dim A,B,C,
 D[3,10]
```

Der Wert D [3,10] enthält Fragezeichen. Alle zweifelhaften Werte lassen sich in folgender Reihenfolge aufrufen:

D[3,10],	D[2,10],	D[1,10],	D[3,9],	D[2,9],
D[1,9],	D[3,8],	D[2,8],	D[1,8],	D[3,7],
D[2,7],	D[1,7],	D[3,6],	D[2,6],	D[1,6],
D[3,5],	D[2,5],	D[1,5],	D[3,4],	D[2,4],
D[1,4],	D[3,3],	D[2,3],	D[1,3],	D[3,2],
D[2,2],	D[1,2],	D[3,1],	D[2,1],	D[1,1],
C,	B			

Fehler im File-Kopf

Wird durch die Fehlermeldung 47 ein Fehler im File-Kopf angezeigt, so wird:

1. der Magnetkopf und der Antriebsmechanismus wie auf Seite 103 gereinigt. Dadurch kann der Fehler behoben sein.

2. Der Befehl, bei dem der Fehler auftrat, wird wiederholt.

3. Tritt der Fehler immer noch auf, so muß der File-Kopf neu markiert werden.

Warnung

Das neue Markieren eines File-Kopfes ist die „letzte Rettung". Wird der File-Kopf neu markiert, so gehen alle Daten und Programme innerhalb dieser File verloren. Darauffolgende Files werden jedoch nicht beeinträchtigt.

Der File-Kopf von File N (die nicht eingelesen werden konnte) wird wie folgt neu markiert:

```
fdf N-1
mrk0,0
```

Positioniert das Band
Markiert den File-Kopf neu

Für File O:

```
rew
mrk0,0
```

Positioniert das Band
Markiert den File-Kopf neu.

Nachdem der File-Kopf neu markiert wurde, beträgt die absolute Größe der File 2 Byte.

Umspulen des Bandes

Wird nur ein kurzes Stück des Bandes (ca. 1 m) oft hintereinander abgefragt, so kann die Bandspannung nachlassen (kann auch bei starken Temperaturveränderungen vorkommen). Dabei kann das Band mit der Kassette in Berührung kommen und dabei beschädigt werden. Im schlimmsten Fall könnte sich das Band einklemmen.

HINWEIS

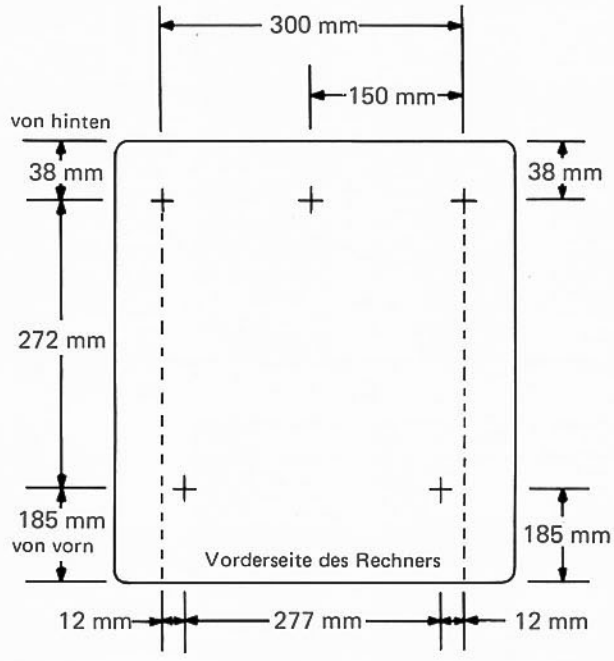
Dieses Problem kann auftreten, wenn fast ausschließlich eine oder zwei Files am Bandanfang oder Bandende verwendet werden.

Ist dies nicht zu umgehen, so sollte das Band in periodischen Abständen vollständig umgespult werden. Beispiel: Werden bei einem Band mit 80 Files nur die Files Ø und 1 verwendet, so sollte das folgende Programm jeweils nach 200 Operationen mit File Ø oder 1 ausgeführt werden.

```
10: fdf80
19: rew
```

ANHANG G
TISCHINSTALLATION

Ihr Rechner kann auf Ihrem Schreibtisch oder Tisch auch festgeschraubt werden. Dazu werden nach der gezeigten Schablone Löcher für 6 bis 32 (National Coarse) Schrauben gebohrt. Die Schrauben sollten etwa 12 mm über die Platte vorstehen.



ADRESSENVERZEICHNIS

HP-VERTRIEBS- UND SERVICEBÜROS

UNITED STATES

ALABAMA
8290 Whitesburg Dr., S.E.
P.O. Box 4207
Huntsville 35802
Tel: (205) 881-4591
TWX: 810-726-2204

Medical Only
228 W. Valley Ave.,
Room 302
Birmingham 35209
Tel: (205) 879-2081/2

ARIZONA
2336 E. Magnolia St.
Phoenix 85034
Tel: (602) 244-1361
TWX: 910-951-1331

2424 East Aragon Rd.
Tucson 85706
Tel: (602) 889-4661

***ARKANSAS**
Medical Service Only
Little Rock 72205
Tel: (501) 664-8773

CALIFORNIA
1430 East Orangethorpe Ave.
Fullerton 92631
Tel: (714) 870-1000
TWX: 910-592-1288

3939 Lankershim Boulevard
North Hollywood 91604
Tel: (213) 877-1282
TWX: 910-499-2170

6305 Arizona Place
Los Angeles 90045
Tel: (213) 649-2511
TWX: 910-328-6147

***Los Angeles**
Tel: (213) 776-7500

3003 Scott Boulevard
Santa Clara 95050
Tel: (408) 249-7000
TWX: 910-338-0518

***Ridgecrest**
Tel: (714) 446-6165

2220 Watt Ave.
Sacramento 95825
Tel: (916) 482-1463
TWX: 910-367-2092

9606 Aero Drive
P.O. Box 23333
San Diego 92123
Tel: (714) 279-3200
TWX: 910-335-2000

Calculators Only
601 California St.
San Francisco 94108
Tel: (415) 989-8470

COLORADO
5600 South Ulster Parkway
Englewood 80110
Tel: (303) 771-3455
TWX: 910-935-0705

CONNECTICUT
12 Lunar Drive
New Haven 06525
Tel: (203) 389-6551
TWX: 710-465-2029

FLORIDA
P.O. Box 24210
2806 W. Oakland Park Blvd.
Ft. Lauderdale 33307
Tel: (305) 731-2020
TWX: 510-955-4099

***Jacksonville**
Medical Service only
Tel: (904) 725-6333

P.O. Box 13910
6177 Lake Ellenor Dr.
Orlando 32809
Tel: (305) 859-2900
TWX: 810-850-0113

21 East Wright St.
Suite 1
Pensacola 32501
Tel: (904) 434-3081

GEORGIA
P.O. Box 28234
450 Interstate North
Atlanta 30328
Tel: (404) 434-4000
TWX: 810-766-4890

HAWAII
2875 So. King Street
Honolulu 96814
Tel: (808) 955-4455

ILLINOIS
5500 Howard Street
Skokie 60076
Tel: (312) 677-0400
TWX: 910-223-3613

***St. Joseph**
Tel: (217) 469-2133

INDIANA
7301 North Shadeland Ave.
Indianapolis 46250
Tel: (317) 842-1000
TWX: 810-260-1796

IOWA
1902 Broadway
Iowa City 52240
Tel: (319) 338-9466
Night: (319) 338-9467

***KANSAS**
Derby
Tel: (316) 267-3655

LOUISIANA
P.O. Box 840
3239 Williams Boulevard
Kenner 70062
Tel: (504) 721-6201
TWX: 810-955-5524

KENTUCKY
Medical/Calculator Only
8003 Troutwood Court
Louisville 40291
Tel: (502) 426-4341

MARYLAND
6707 Whitestone Road
Baltimore 21207
Tel: (301) 944-5400
TWX: 710-862-9157

4 Choke Cherry Road
Rockville 20850
Tel: (301) 948-6370
TWX: 710-828-9685
710-828-0487

P.O. Box 1648
2 Choke Cherry Road
Rockville 20850
Tel: (301) 948-6370
TWX: 710-828-9684

MASSACHUSETTS
32 Hartwell Ave.
Lexington 02173
Tel: (617) 861-8960
TWX: 710-326-6904

MICHIGAN
23855 Research Drive
Farmington 48024
Tel: (313) 476-6400
TWX: 810-242-2900

MINNESOTA
2400 N. Prior Ave.
Roseville 55113
Tel: (612) 636-0700
TWX: 910-563-3734

MISSISSIPPI
***Jackson**
Medical Service only
Tel: (601) 982-9363

MISSOURI
11131 Colorado Ave.
Kansas City 64137
Tel: (816) 763-8000
TWX: 910-771-2087

148 Weldon Parkway
Maryland Heights 63043
Tel: (314) 567-1455
TWX: 910-764-0830

NEBRASKA
Medical Only
11902 Elm Street
Suite 4C
Omaha 68144
Tel: (402) 333-6017

NEW JERSEY
W. 120 Century Rd.
Paramus 07652
Tel: (201) 265-5000
TWX: 710-990-4951

NEW MEXICO
P.O. Box 11634
Station E
11300 Lomas Blvd., N.E.
Albuquerque 87123
Tel: (505) 292-1330
TWX: 910-989-1185

156 Wyatt Drive
Las Cruces 88001
Tel: (505) 526-2485
TWX: 910-983-0550

NEW YORK
6 Automation Lane
Computer Park
Albany 12205
Tel: (518) 458-1550
TWX: 710-441-8270

Calculators Only
1251 Avenue of the Americas
Floor 32 - Suite 3296
New York City 10020
Tel: (212) 265-5575

New York City
Manhattan, Bronx
Contact Paramus, NJ Office
Tel: (201) 265-5000
Brooklyn, Queens, Richmond
Contact Woodbury, NY Office
Tel: (516) 921-0300

201 South Avenue
Poughkeepsie 12601
Tel: (914) 454-7330
TWX: 510-248-0012

39 Saginaw Drive
Rochester 14623
Tel: (716) 473-9500
TWX: 510-253-5981

5858 East Molloy Road
Syracuse 13211
Tel: (315) 455-2486
TWX: 710-541-0482

1 Crossways Park West
Woodbury 11797
Tel: (516) 921-0300
TWX: 510-221-2168

NORTH CAROLINA
P.O. Box 5188
1923 North Main Street
High Point 27262
Tel: (919) 885-8101
TWX: 510-926-1516

OHIO
16500 Sprague Road
Cleveland 44130
Tel: (216) 243-7300
Night: 243-7305
TWX: 810-423-9431

330 Progress Rd.
Dayton 45449
Tel: (513) 859-8202
TWX: 810-459-1925

1041 Kingsmill Parkway
Columbus 43229
Tel: (614) 436-1041

OKLAHOMA
P.O. Box 32008
Oklahoma City 73132
Tel: (405) 721-0200
TWX: 910-830-6862

OREGON
17890 SW Boones Ferry Road
Tualatin 97062
Tel: (503) 620-3350
TWX: 910-467-8714

PENNSYLVANIA
111 Zeta Drive
Pittsburgh 15238
Tel: (412) 782-0400
Night: 782-0401
TWX: 710-795-3124

1021 8th Avenue
King of Prussia Industrial Park
King of Prussia 19406
Tel: (215) 265-7000
TWX: 510-660-2670

SOUTH CAROLINA
6941-0 N. Trenholm Road
Columbia 29260
Tel: (803) 782-6493

TENNESSEE
***Memphis**
Medical Service only
Tel: (901) 274-7472

***Nashville**
Medical Service only
Tel: (615) 244-5448

TEXAS
P.O. Box 1270
201 E. Arapaho Rd.
Richardson 75080
Tel: (214) 231-6101
TWX: 910-867-4723

P.O. Box 27409
6300 Westpark Drive
Suite 100
Houston 77027
Tel: (713) 781-6000
TWX: 910-881-2645

205 Billy Mitchell Road
San Antonio 78226
Tel: (512) 434-8241
TWX: 910-871-1170

UTAH
2890 South Main Street
Salt Lake City 84115
Tel: (801) 487-0715
TWX: 910-925-5681

VIRGINIA
Medical Only
P.O. Box 12778
No. 7 Koger Exec. Center
Suite 212
Norfolk 23502
Tel: (804) 497-1026/7

P.O. Box 9854
2914 Hungary Springs Road
Richmond 23228
Tel: (804) 285-3431
TWX: 710-956-0157

WASHINGTON
Bellefield Office Pk.
1203-114th SE
Bellevue 98004
Tel: (206) 454-3971
TWX: 910-443-2446

***WEST VIRGINIA**
Medical/Analytical Only
Charleston
Tel: (304) 345-1640

WISCONSIN
9431 W. Beloit Road
Suite 117
Milwaukee 53227
Tel: (414) 541-0550

FOR U.S. AREAS NOT LISTED:
Contact the regional office nearest you: Atlanta, Georgia; North Hollywood, California; Rockville, (4 Choke Cherry Rd.) Maryland; Skokie, Illinois. Their complete addresses are listed above.

***Service Only**

CANADA

ALBERTA
Hewlett-Packard (Canada) Ltd.
11748 Kingsway Ave.
Edmonton T5G 0X5
Tel: (403) 452-3670
TWX: 610-831-2431

Hewlett-Packard (Canada) Ltd.
915-42 Avenue S.E. Suite 102
Calgary T2G 1Z1
Tel: (403) 287-1672

BRITISH COLUMBIA
Hewlett-Packard (Canada) Ltd.
837 E. Cordova Street
Vancouver V6A 3R2
Tel: (604) 254-0531
TWX: 610-922-5059

MANITOBA
Hewlett-Packard (Canada) Ltd.
513 Century St.
St. James
Winnipeg R3H 0L8
Tel: (204) 786-7581
TWX: 610-671-3531

NOVA SCOTIA
Hewlett-Packard (Canada) Ltd.
800 Windmill Road
Dartmouth B3C 1L1
Tel: (902) 469-7820

ONTARIO
Hewlett-Packard (Canada) Ltd.
1785 Woodward Dr.
Ottawa K2C 0P9
Tel: (613) 225-6530
TWX: 610-562-8968

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
Mississauga L4V 1L9
Tel: (416) 678-9430
TWX: 610-492-4246

QUEBEC
Hewlett-Packard (Canada) Ltd.
275 Hymus Blvd.
Pointe Claire H9R 1G7
Tel: (514) 697-4232
TWX: 610-422-3022
TLX: 05-821521 HPCL

Hewlett-Packard (Canada) Ltd.
2376 Galvani Street
Ste-Foy G1N 4G4
Tel: (418) 688-8710

FOR CANADIAN AREAS NOT LISTED:
Contact Hewlett-Packard (Canada) Ltd. in Mississauga.

CENTRAL AND SOUTH AMERICA

ARGENTINA
Hewlett-Packard Argentina
S.A.C.e.I.
Lavalle 1171-3° Piso
Buenos Aires
Tel: 35-0436, 35-0627, 35-0341
Telex: 012-1009
Cable: HEWPACK ARG

BOLIVIA
Stambuk & Mark (Bolivia) Ltda.
Av. Mariscal Santa Cruz 1342
La Paz
Tel: 40626, 53163, 52421
Telex: 3560014
Cable: BUKMAR

BRAZIL
Hewlett-Packard Do Brasil
I.E.C. Ltda.
Rua Frei Caneca, 1 152-Bela Vista
01307-**São Paulo**-SP
Tel: 288-71-11, 287-81-20
287-61-93
Telex: 309151 2 3
Cable: HEWPACK São Paulo

Hewlett-Packard Do Brasil
I.E.C. Ltda.
Praca Dom Feliciano, 78-8° andar (Sala 806/8)
9000-**Pôrto Alegre**-RS
Tel: 25-84-70-DDD (0512)
Cable: HEWPACK Pôrto Alegre

Hewlett-Packard Do Brasil
I.E.C. Ltda.
Rua Siqueira Campos, 53, 4° andar Copacabana
2000-**Rio de Janeiro**-GB
Tel: 257-80-94-DDD (021)
Telex: 2100 79 HEWPACK
Cable: HEWPACK
Rio de Janeiro

CHILE
Calcagni y Metcalfe Ltda.
Calle Lira 81, Oficina 5
Casilla 2118
Santiago, 1
Tel: 398613
Cable: CALMET

COLOMBIA
Instrumentación
Henrik A. Langebaek & Kier S.A.
Carrera 7 No. 48-59
Apartado Aéreo 6287
Bogota, 1 D.E.
Tel: 45-78-06, 45-55-46
Cable: AARIS Bogota
Telex: 44400INSTCO

COSTA RICA
Cientifica Costarricense S.A.
Apartado 10159
San José
Tel: 21-86-13
Cable: GALGUR San José

GUATEMALA
IPESA
Avenida La Reforma 3-48, Zona 9
Guatemala
Tel: 63627, 64786
Telex: 4192 TELTRO GU

MEXICO
Hewlett-Packard Mexicana, S.A. de C.V.
Torres Adalid No. 21, 11° Piso
Col. del Valle
Mexico 12, D.F.
Tel: (905) 543-42-32
Telex: 017-74-507

Hewlett-Packard Mexicana, S.A. de C.V.
Ave. Constitución No. 2184
Monterrey, N.L.
Tel: 48-71-32, 48-71-84

NICARAGUA
Roberto Terán G.
Apartado Postal 689
Edificio Terán
Managua
Tel: 3451, 3452
Cable: ROTERAN Managua

PANAMA
Electrónico Balboa, S.A.
P.O. Box 4929
Calle Samuel Lewis
Ciudad de Panama
Tel: 64-2700
Telex: 3431103 Curunda, Canal Zone
Cable: ELECTRON Panama

PARAGUAY
Z.J. Melamed S.R.L.
Division: Aparatos y Equipos Medicos
Division: Aparatos y Equipos Scientificos y de Investigacion
P.O. Box 676
Chile, 482, Edificio Victoria
Asunción
Tel: 4-5069, 4-6272
Cable: RAMEL

PERU
Compañia Electro Médica S.A.
Ave. Enrique Canaval 312
San Isidro
Casilla 1030
Lima
Tel: 22-3900
Cable: ELMED Lima

PUERTO RICO
San Juan Electronics, Inc.
P.O. Box 5167
Ponce de León 154
Pda. 3-PTA de Tierra
San Juan 00906
Tel: (809) 725-3342, 722-3342
Cable: SATRONICS San Juan
Telex: SATRON 3450 332

URUGUAY
Pablo Ferrando S.A.
Comercial e Industrial
Avenida Italia 2877
Casilla de Correo 370
Montevideo
Tel: 40-3102
Cable: RADIUM Montevideo

VENEZUELA
Hewlett-Packard de Venezuela C.A.
Apartado 50933
Edificio Segre
Tercera Transversal
Los Ruices Norte
Caracas 107
Tel: 35-00-11
Telex: 21146 HEWPACK
Cable: HEWPACK Caracas

FOR AREAS NOT LISTED, CONTACT:
Hewlett-Packard
Inter-Americas
3200 Hillview Ave.
Palo Alto, California 94304
Tel: (415) 493-1501
TWX: 910-373-1260
Cable: HEWPACK Palo Alto
Telex: 034-8300 034-8493

EUROPE

AUSTRIA
Hewlett-Packard Ges.m.b.H.
Handelska 52/3
P.O. Box 7
A-1205 **Vienna**
Tel: (0222) 33 66 06 to 09
Cable: HEWPAK Vienna
Telex: 75923 hewpak a

BELGIUM
Hewlett-Packard Benelux
S.A./N.V.
Avenue de Col-Vert, 1,
(Groenkraaglaan)
B-1170 **Brussels**
Tel: (02) 672 22 40
Cable: PALOBEN Brussels
Telex: 23 494 paloben bru

DENMARK
Hewlett-Packard A/S
Datavej 52
DK-3460 **Birkerød**
Tel: (01) 81 66 40
Cable: HEWPACK AS
Telex: 166 40 hp as

Hewlett-Packard A/S
Navervej 1
DK-8600 **Silkeborg**
Tel: (06) 82 71 66
Telex: 166 40 hp as
Cable: HEWPACK AS

FINLAND
Hewlett-Packard Oy
Nahkahousuntie 5
P.O. Box 6
SF-00211 **Helsinki** 21
Tel: 6923031
Cable: HEWPACKOY Helsinki
Telex: 12-15363

FRANCE
Hewlett-Packard France
Quartier de Courtaboeuf
Boite Postale No. 6
F-91401 **Orsay**
Tel: (1) 907 78 25
Cable: HEWPACK Orsay
Telex: 60048

Hewlett-Packard France
Agence Régional
Chemin des Mouilles
Boite Postale No. 12
F-69130 **Ecully**
Tel: (78) 33 81 25.
Cable: HEWPACK Ecully
Telex: 31 617

Hewlett-Packard France
Agence Régionale
Zone Aéronautique
Avenue Clément Ader
F-31770 **Colomiers**
Tel: (61) 78 11 55
Telex: 51957

Hewlett-Packard France
Agence Régionale
Centre d'aviation générale
F-13721 **Aéroport de Marignane**
Tel: (91) 89 12 36
TWX: 41770 F

Hewlett-Packard France
Agence Régionale
63, Avenue de Rochester
F-35014 **Rennes**
Tel: 74912 F
Telex: 74 912 F

Hewlett-Packard France
Agence Régionale
74, Allée de la Robertsau
F-67000 **Strasbourg**
Tel: (88) 35 23 20/21
Telex: 89141
Cable: HEWPACK STRBG

Medical/Calculator Only
Hewlett-Packard France
Agence Régionale
Centre Vauban
201, rue Colbert
Entrée Az
F-59000 **Lille**
Tel: (20) 51 44 14

GERMAN FEDERAL REPUBLIC
Hewlett-Packard GmbH
Vertriebszentrale Frankfurt
Bernerstrasse 117
Postfach 560 140
D-6000 **Frankfurt** 56
Tel: (0611) 50 04-1
Cable: HEWPACKSA Frankfurt
Telex: 41 32 49 fra

Hewlett-Packard GmbH
Technisches Buero Böblingen
Herrenbergerstrasse 110
D-7030 **Böblingen**, Württemberg
Tel: (07031) 66 72 87
Cable: HEPAK Böblingen
Telex: 72 65 739 bbn

Hewlett-Packard GmbH
Technisches Buero Düsseldorf
Vogelsanger Weg 38
D-4000 **Düsseldorf**
Tel: (0211) 63 80 31/5
Telex: 85/86 533 hpdd d

Hewlett-Packard GmbH
Technisches Buero Hamburg
Wendenstrasse 23
D-2000 **Hamburg** 1
Tel: (040) 24 13 93
Cable: HEWPACKSA Hamburg
Telex: 21 63 032 hphh d

Hewlett-Packard GmbH
Technisches Buero Hannover
Mellendorfer Strasse 3
D-3000 **Hannover-Kleefeld**
Tel: (0511) 55 60 46
Telex: 092 3259

Hewlett-Packard GmbH
Technisches Buero Nuremberg
Hersbruckerstrasse 42
D-8500 **Nuremberg**
Tel: (0911) 57 10 66
Telex: 623 860

Hewlett-Packard GmbH
Technisches Buero München
Unterhachinger Strasse 28
ISAR Center
D-8012 **Ottobrunn**
Tel: (089) 601 30 61/7
Telex: 52 49 85
Cable: HEWPACKSA München

(West Berlin)
Hewlett-Packard GmbH
Technisches Buero Berlin
Keith Strasse 2-4
D-1000 **Berlin** 30
Tel: (030) 24 90 86
Telex: 18 34 05 hpbln d

GREECE
Kostas Karayannis
18, Ermou Street
GR-**Athens** 126
Tel: 3230-303 Sales/SVC
3230-305 Adm. Order Proc.
Cable: RAKAR Athens
Telex: 21 59 62 rkar gr

Hewlett-Packard S.A.
Mediterranean & Middle East
Operations
35 Kolokotroni Street
Platia Kefallariou
Gr-Kifissia-**Athens**
Tel: 8080337, 8080359,
8080429, 8018693
Telex: 21 6588
Cable: HEWPACKSA Athens

Analytical Only
"INTECO" G. Papathanassiou & Co.
Marni 17
GR - **Athens** 103
Tel: 521 915
Cable: INTEKNIKA
Telex: 21 5329 INTE GR

Medical Only
Technomed Hellas Ltd.
52, Skoufa Street
GR - **Athens** 135
Tel: 626 972
Cable: ETALAK Athens
Telex: 21-4693 ETAL GR

IRELAND
Hewlett-Packard Ltd.
King Street Lane
Winnersh, Wokingham
GB-**Berkshire** RG11 5AR
Tel: Wokingham 784774
Telex: 847178/848179

Hewlett-Packard Ltd.
"The Graftons"
Stamford New Road
GB-**Altrincham**, Cheshire
Tel: (061) 928-9021
Telex: 668068

ITALY
Hewlett-Packard Italiana S.p.A.
Via Amerigo Vespucci 2
I-20124 **Milan**
Tel: (2) 6251 (10 lines)
Cable: HEWPACKIT Milan
Telex: 32046

Hewlett-Packard Italiana S.p.A.
Via Pietro Maroncelli 40
(ang. Via Visentin)
I-35100 **Padova**
Tel: 66 40 62/66 31 88
Telex: 32046 via Milan

Medical Only
Hewlett-Packard Italiana S.p.A.
Via Medaglie d'Oro, 2
I-56100 **Pisa**
Tel: (050) 2 32 04
Telex: 32046 via Milan

Hewlett-Packard S.p.A.
Via G. Armellini 10
I-00143 **Rome-Eur**
Tel: (6) 5912544/5
Telex: 61514
Cable: HEWPACKIT Rome

Hewlett-Packard Italiana S.p.A.
Via San Quintino, 46
I-10121 **Turin**
Tel: 53 82 64/54 84 68
Telex: 32046 via Milan

Medical/Calculators Only
Hewlett-Packard Italiana S.p.A.
Via Principe Nicola 43 G/C
I-95126 **Catania**
Tel: (095) 370505

LUXEMBURG
Hewlett-Packard Benelux
S.A./N.V.
Avenue de Col-Vert, 1,
(Groenkraaglaan)
B-1170 **Brussels**
Tel: (02) 672 22 40
Cable: PALOBEN Brussels
Telex: 23 494

NETHERLANDS
Hewlett-Packard Benelux N.V.
Weerdestein 117
P.O. Box 7825
NL-**Amsterdam**, 1011
Tel: (020) 5411522
Cable: PALOBEN Amsterdam
Telex: 13 216 hepa nl

NORWAY
Hewlett-Packard Norge A/S
Nesveien 13
Box 149
N-1344 **Haslum**
Tel: (02) 53 83 60
Telex: 16621 hpnas n

POLAND
Analytical/Medical Only
Hewlett-Packard
Warsaw Technical Office
Szpitalna 1
00-120 **Warsaw**
Tel: 268031
Telex: 812453

PORTUGAL
Telectra-Empresa Técnica de
Equipamentos Eléctricos S.a.r.l.
Rua Rodrigo da Fonseca 103
P.O. Box 2531
P-**Lisbon** 1
Tel: (19) 68 60 72
Cable: TELECTRA Lisbon
Telex: 12598

Mundinter
Intercambio Mundial de Comércio
Sarl Avenida Antonio Augusto
de Aguiar 138
P.O. Box 2761
P - **Lisbon**
Tel: (19) 53 21 31/7
Cable: INTERCAMBIO Lisbon

SPAIN
Hewlett-Packard Española, S.A.
Jerez No. 3
E-**Madrid** 16
Tel: (1) 458 26 00 (10 lines)
Telex: 23515 hpe

Hewlett-Packard Española, S.A.
Milanesado 21-23
E-**Barcelona** 17
Tel: (3) 203 6200 (5 lines)
Telex: 52603 hpbe e

Hewlett-Packard Española, S.A.
Av Ramon y Cajal, 1
Edificio Sevilla I, planta 9°
E-**Seville**
Tel: 64 44 54/58

Hewlett-Packard Española S.A.
Edificio Albia II 7° B
E-**Bilbao**
Tel: 23 83 06/23 82 06

Calculators Only
Hewlett-Packard Española S.A.
Alvaro Bazen, 12
(Edificio Luz)
E - **Valencia** - 10
Tel: 60 42 00

SWEDEN
Hewlett-Packard Sverige AB
Enighetsvägen 1-3
Fack
S-161 20 **Bromma** 20
Tel: (08) 730 05 50
Cable: MEASUREMENTS
Stockholm
Telex: 10721

Hewlett-Packard Sverige AB
Hagakersgatan 9C
S-431 41 **Mölndal**
Tel: (031) 27 68 00/01
Telex: Via Bromma

SWITZERLAND
Hilett-Packard (Schweiz) AG
Zurcherstrasse 20
P.O. Box 64
CH-8952 Schlieren **Zurich**
Tel: (01) 98 18 21
Cable: HPAG CH
Telex: 53933 hpag

Hewlett-Packard (Schweiz) AG
9, chemin Louis-Pictet
CH-1214 Vernier-**Geneva**
Tel: (022) 41 49 50
Cable: HEWPACKSA Geneva
Telex: 27 333 hpsa ch

TURKEY
Telekom Engineering Bureau
Saglik Sok No. 15/1
Ayaspasa-Beyoglu
P.O. Box 437 Beyoglu
TR-**Istanbul**
Tel: 49 40 40
Cable: TELEMATION Istanbul

UNITED KINGDOM
Hewlett-Packard Ltd.
King Street Lane
Winnersh, Wokingham
GB-**Berkshire** RG11 5AR
Tel: Wokingham 784774
Telex: 847178/848179

Hewlett-Packard Ltd.
"The Graftons"
Stamford New Road
GB-**Altrincham**, Cheshire
Tel: (061) 928-9021
Telex: 668068

Hewlett-Packard Ltd
c/o Makro
South Service Wholesale Centre
Amber Way
Halesowen Industrial Estate
GB-**Halesowen**, Worcs
Tel: Birmingham 7860

Hewlett-Packard Ltd
4th Floor
Wedge House
799, London Road
GB-**Thornton Heath** CR4 6XL,
Surrey
Tel: (01) 684 0105
Telex: 946825

Hewlett-Packard Ltd
c/o Makro
South Service Wholesale Centre
Wear Industrial Estate
Washington
GB-**New Town**, County Durham
Tel: Washington 464001 ext. 57/58

Hewlett-Packard Ltd.'s registered
address for V.A.T. purposes
only:
70, Finsbury Pavement
London, EC2A1SX
Registered No. 690597

USSR
Hewlett-Packard USSR
c/o Commercial Office
American Embassy (Box M)
A-1091 **Vienna**, Austria
Tel: 221-79-71
Telex: 7825 hewpak SU

YUGOSLAVIA
Iskra-Standard/Hewlett-Packard
Topniska 58/3
61000 **Ljubljana**
Tel: 315-879/321-674
Telex: 31300

SOCIALIST COUNTRIES PLEASE CONTACT:
Hewlett-Packard S.A.
7, rue du Bois-du-Lan
P.O. Box 349
CH-1217 Meyrin 1 **Geneva**
Switzerland
Tel: (022) 41 54 00
Cable: HEWPACKSA Geneva
Telex: 2 24 86

AFRICA, ASIA, AUSTRALIA

ANGOLA
Telectra
Empresa Técnica de
Equipamentos
Eléctricos, S.A.R.L.
R. Barbosa Rodrigues, 42-1°DT.°
Caixa Postal, 6487-**Luanda**
Tel: 35515/6
Cable: TELECTRA Luanda

AUSTRALIA
Hewlett-Packard Australia
Pty. Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
Tel: 89-6351, 89-6306
Telex: 31-024
Cable: HEWPARD Melbourne

Hewlett-Packard Australia
Pty. Ltd.
31 Bridge Street
Pymble,
New South Wales, 2073
Tel: 449-6566
Telex: 21561
Cable: HEWPARD Sydney

Hewlett-Packard Australia
Pty. Ltd.
97 Churchill Road
Prospect 5082
South Australia
Tel: 44 8151
Cable: HEWPARD Adelaide

Hewlett-Packard Australia
Pty. Ltd.
141 Stirling Highway
Claremont, W.A. 6010
Tel: 86-5455
Telex: 93859
Cable: HEWPARD

Hewlett-Packard Australia
Pty. Ltd.
121 Wollongong Street
Fyshwick, A.C.T., 2609
Tel: 95 3733

Hewlett-Packard Australia
Pty. Ltd.
5th Floor
Teachers Union Building
495-499 Boundary Street
Spring Hill, 4000 Queensland
Tel: 29 1544
Telex: AA-42133

CEYLON
United Electricals Ltd
P.O. Box 681
60, Park St.
Colombo 2
Tel: 26696
Cable: HOTPOINT Colombo

CYPRUS
Kypronics
19 Gregorios & Xenopoulos Rd
P.O. Box 1152
CY-**Nicosia**
Tel: 45628 29
Cable: KYPRONICS PANDEHIS

HONG KONG
Schmidt & Co (Hong Kong) Ltd.
P.O. Box 297
Connaught Centre
39th Floor
Connaught Road, Central
Hong Kong
Tel: 240168, 232735
Telex: HX4766
Cable: SCHMIDTCO Hong Kong

INDIA
Blue Star Ltd.
Kasturi Buildings
Jamshedji Tata Rd.
Bombay 400 020
Tel: 29 50 21
Telex: 3751
Cable: BLUEFROST

Blue Star Ltd.
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
Bombay 400 025
Tel: 45 78 87
Telex: 4093
Cable: FROSTBLUE

Blue Star Ltd.
Band Box House
Prabhadevi
Bombay 400 025
Tel: 45 73 01
Telex: 3751
Cable: BLUESTAR

Blue Star Ltd.
14/40 Civil Lines
Kanpur 208 001
Tel: 6 88 82
Cable: BLUESTAR

Blue Star Ltd.
7 Hare Street
P.O. Box 506
Calcutta 700 001
Tel: 23-0131
Telex: 655
Cable: BLUESTAR

Blue Star Ltd.
Blue Star House,
34 Ring Road
Lajpat Nagar
New Delhi 110 024
Tel: 62 32 76
Telex: 2463
Cable: BLUESTAR

Blue Star Ltd.
Blue Star House
11 11A Magarath Road
Bangalore 560 025
Tel: 55668
Telex: 430
Cable: BLUESTAR

Blue Star Ltd.
Meenakshi Mandiran
xxx 1678 Mahatma Gandhi Rd
Cochin 682 016 Kerala

Blue Star Ltd.
1-1-117/1
Sarojini Devi Road
Secunderabad 500 003
Tel: 7 63 91, 7 73 93
Cable: BLUEFROST
Telex: 459

Blue Star Ltd.
23/24 Second Line Beach
Madras 600 001
Tel: 23954
Telex: 379
Cable: BLUESTAR

Blue Star Ltd.
Nathraj Mansions
2nd Floor Bistupur
Jamshedpur 831 001
Tel: 38 04
Cable: BLUESTAR
Telex: 240

INDONESIA
BERCA Indonesia P.T.
P.O. Box 496
1st Floor JL. Cikini Raya 61
Jakarta
Tel: 56038, 40369, 49886
Telex: 2895 Jakarta

IRAN
Multi Corp International Ltd
Avenue Soraya 130
P.O. Box 1212
IR-**Teheran**
Tel: 83 10 35-39
Cable: MULTICORP Tehran
Telex: 2893 mci ir

ISRAEL
Electronics & Engineering
Div. of Motorola Israel Ltd.
17 Aminadav Street
Tel-Aviv
Tel: 36941 (3 lines)
Cable: BASTEL Tel-Aviv
Telex: 33569

JAPAN
Yokogawa-Hewlett-Packard Ltd
Ohashi Building
1-59-1 Yoyogi
Shibuya-ku, **Tokyo**
Tel: 03-370-2281/92
Telex: 232-2024YHP
Cable: YHPMARKET TOK 23-724

Yokogawa-Hewlett-Packard Ltd
Nisei Ibaragi Bldg
2-2-8 Kasuga
Ibaragi-Shi
Osaka
Tel: (0726) 23-1641
Telex: 5332-385 YHP OSAKA

Yokogawa-Hewlett-Packard Ltd
Nakamo Building
No. 24 Kamisasazma-cho
Nakamura-ku, **Nagoya** City
Tel: (052) 571-5171

Yokogawa-Hewlett-Packard Ltd.
Tamigawa Building
2-24-1 Tsuruya-cho
Kanagawa-ku
Yokohama, 221
Tel: 045-312-1252
Telex: 382-3204 YHP YOK

Yokogawa-Hewlett-Packard Ltd.
Mito Mitsui Building
1-4-73 San-no-maru
Mito, 310
Tel: 0292-25-7470

Yokogawa-Hewlett-Packard Ltd.
Inoue Building
1348-3, Asahi-cho, 1-chome
Atsugi, 243
Tel: 0462-24-0452

KENYA
Technical Engineering Services
P.O. Box 18311
Nairobi, Kenya
Tel: 57726
Cable: PROTON

KOREA
American Trading Company
Korea
I.P.O. Box 1103
Dae Kyung Bldg., 8th Floor
107 Sejong-Ro,
Chongro-Ku, **Seoul**
Tel: (4 lines) 73-8924-7
Cable: AMTRACO Seoul

KUWAIT
Al-Khaldiya Trading &
Contracting Co.
Al Soor Street
Michaan Bldg. No. 4
Kuwait
Tel: 42 99 10
Cable: VISCOUNT

LEBANON
Constantin E. Macridis
Clemenceau Street 34
P.O. Box 7213
RL-**Beirut**
Tel: 220846
Telex: 21114 Leb
Cable: ELECTRONUCLEAR Beirut

MALAYSIA
MECOMB Malaysia Ltd
2 Lorong 13/6A
Section 13
Petaling Jaya, **Selangor**
Cable: MECOMB Kuala Lumpur

MOZAMBIQUE
A.N. Goncalves, Lta
162, 1 Apt. 14 Av. D. Luis
Caixa Postal 107
Lourenco Marques
Tel: 27091, 27114
Telex: 6-203 Negon Mo
Cable: NEGON

NEW ZEALAND
Hewlett-Packard (N.Z.) Ltd.
94-96 Dixon Street
P.O. Box 9443
Courtenay Place,
Wellington
Tel: 59-559
Telex: 3898
Cable: HEWPACK Wellington

Hewlett-Packard (N.Z.) Ltd.
Pakuranga Professional Centre
267 Pakuranga Highway
Box 51092
Pakuranga
Tel: 569-651
Cable: HEWPACK, Auckland

Analytical/Medical Only
Dental & Medical Supply Co. Ltd.
Scientific Division
79 Carlton Gore Road
Newmarket
P.O. Box 1234
Auckland
Tel: 75-289
Cable: DENTAL Auckland

NIGERIA
The Electronics
Instrumentations Ltd.
N6B/770 Oyo Road
Oluseun House
P.M.B. 5402
Ibadan
Tel: 22325
Cable: THETEIL Ibadan

The Electronics Instrumenta-
tions Ltd. (TEIL)
16th Floor Cocoa House
P.M.B. 5402
Ibadan
Tel: 22325
Cable: THETEIL Ibadan

PAKISTAN
Mushko & Company, Ltd
Oosman Chambers
Abdullah Haroon Road
Karachi 3
Tel: 511027, 512927
Cable: COOPERATOR Karachi

Mushko & Company, Ltd
38B, Satellite Town
Rawalpindi
Tel: 41924
Cable: FEMUS Rawalpindi

PHILIPPINES
Electromex, Inc.
6th Floor, Amalgamated
Development Corp. Bldg.
Ayala Avenue, Makati, Rizal
C.C.P.O. Box 1028
Makati, Rizal
Tel: 86-18-87, 87-76-77
Cable: ELEMEX Manila

SINGAPORE
Mechanical & Combustion
Engineering Company Pte
Ltd.
10/12, Jalan Kilang
Red Hill Industrial Estate
Singapore, 3
Tel: 647151 (7 lines)
Cable: MECOMB Singapore

Hewlett-Packard Singapore
(Pte.) Ltd.
Blk 2, 6th FLOOR, Jalan
Bukit Merah
Redhill Industrial Estate
Alexandra P.O. Box 87,
Singapore 3
Tel: 633022
Telex: HPSG RS 21486
Cable: HEWPACK, Singapore

SOUTH AFRICA
Hewlett-Packard South Africa
(Pty.), Ltd.
Hewlett-Packard House
Daphne Street, Wendywood
Sandton, Transvaal 2001
Tel: 802-1040
Telex: SA43-4782JH
Cable: HEWPACK

Hewlett-Packard South Africa
(Pty.), Ltd.
Breecastle House
Bree Street
Cape Town
Tel: 2-6941/2/3
Cable: HEWPACK Cape Town
Telex: 0006 CT

Hewlett-Packard South Africa
(Pty.), Ltd.
641 Ridge Road, Durban
P.O. Box 37099
Overport, Durban, 4067
Tel: 88-6102
Telex: 6-7954

TAIWAN
Hewlett-Packard Taiwan
39 Chung Shiao West Road
Sec. 1 Overseas Insurance
Corp. Bldg. 7th Floor
Taipei
Tel: 389160, 1, 2,
Telex: TP824 HEWPACK
Cable: HEWPACK Taipei

Hewlett-Packard Taiwan
38, Pb-Ai Lane, San Min Chu
Kaohsiung
Tel: 297319

THAILAND
UNIMESA Co., Ltd.
Elsom Research Building
Bangjak Sukumvit Ave.
Bangkok
Tel: 932387, 930338
Cable: UNIMESA Bangkok

UGANDA
Uganda Tele-Electric Co., Ltd
P.O. Box 4449
Kampala
Tel: 57279
Cable: COMCO Kampala

VIETNAM
Peninsular Trading Inc
P.O. Box H-3
216 Hien-Vuong
Saigon
Tel: 20-805, 93398
Cable: PENTRA, SAIGON 242

ZAMBIA
R.J. Tilbury (Zambia) Ltd
P.O. Box 2792
Lusaka
Zambia, Central Africa
Tel: 73793
Cable: ARJAYTEE, Lusaka

MEDITERRANEAN AND MIDDLE EAST COUNTRIES NOT SHOWN PLEASE CONTACT:
Hewlett-Packard S.A.
Mediterranean and Middle
East Operations
35, Kolokotroni Street
Platia Kefallariou
GR-Kifissia-**Athens**
Telex: 21-6588
Cable: HEWPACKSA Athens

OTHER AREAS NOT LISTED, CONTACT:
Hewlett-Packard
Export Trade Company
3200 Hillview Ave.
Palo Alto, California 94304
Tel: (415) 493-1501
TWX: 910 373-1267
Cable: HEWPACK Palo Alto
Telex: 034-8300, 034-8493

INDEX

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V W

X Z

HEWLETT hp PACKARD

09825-90095 (German)

Printed in Federal Republic of Germany 6/77