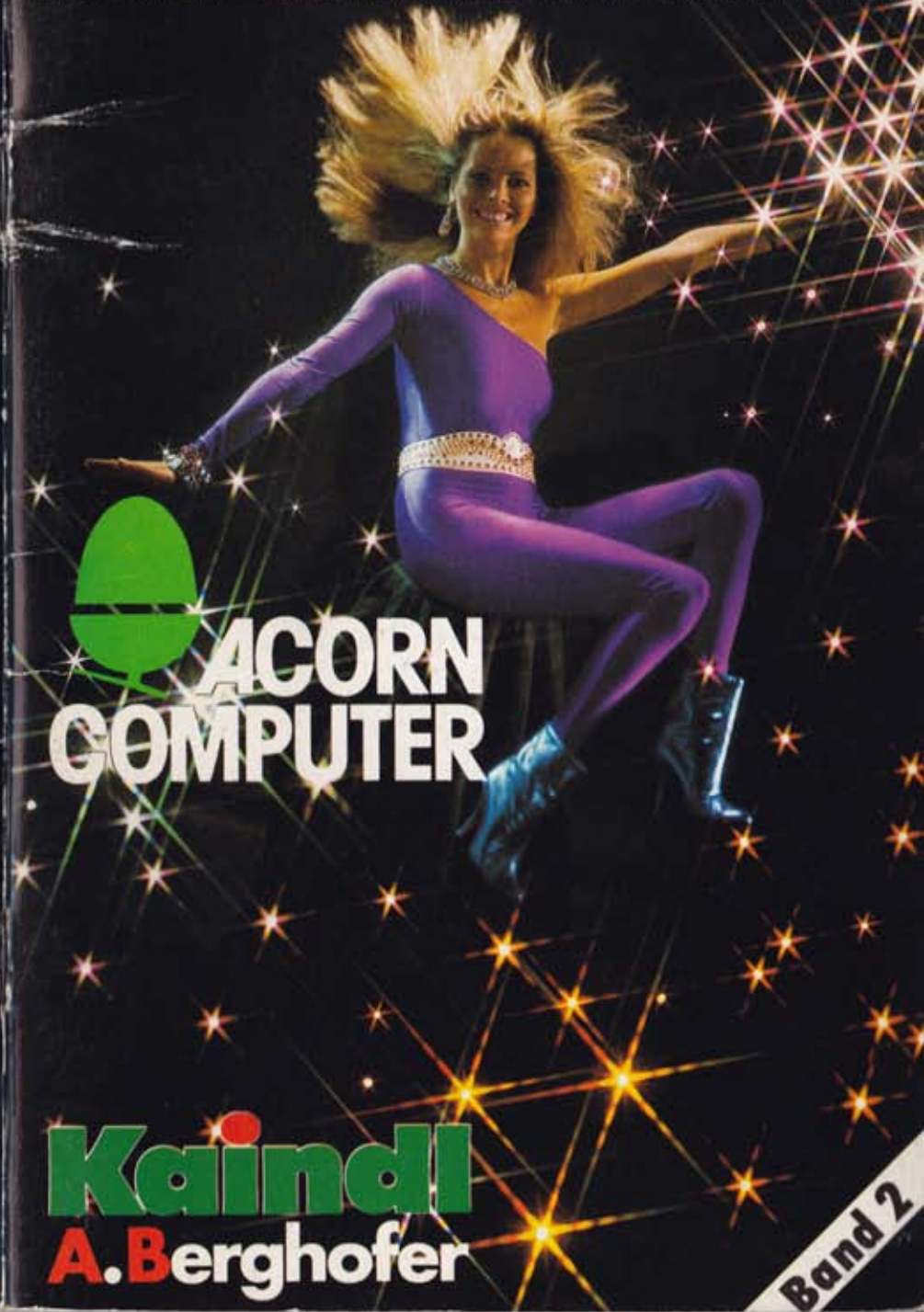


WZUW



 **ACORN  
COMPUTER**

**Kaindi**  
**A. Berghofer**

**Band 2**

**Band 2  
zur ORF-  
Sendereihe  
"COMPUTERFAMILIE"**

**für die Folgen 6-13**

11. Jänner 1985  
18. Jänner 1985  
25. Jänner 1985  
1. Februar 1985  
15. Februar 1985  
22. Februar 1985  
1. März 1985  
8. März 1985

## „WIR NEHMEN AB“

Hallo Computerfreunde! Ich sag' Euch, mein COMPUTER ist schnell – gigantisch! Wenn ich nur annähernd so beweglich wäre wie ER! Aber bei meinem Umfang? Unmöglich! Da müßte ich schon einige Kilo's wegschwitzen. Oder streng nach Kalorien leben. Den ganzen Tag kleinweise aufschreiben, was gefuttert und geschlurft wurde. Mühsal, Mühsal! Und ER, mein COMPUTER? Schlanke Züge, keine Jahresringe! Wie hält ER sich nur fit? Na sicher mit einem SOFTWARE-Paket, einem Programm. Ob ER mir's zeigt? Ich versuch's! Ich tippe LOAD "JOULE" – das soll IHM sein Geheimnis entlocken, hab' ich gehört. Sein Laufwerk schnurrt leise und nach einiger Zeit meldet ER sich wieder mit seinem PROMPT – Zeichen <. Damit will ER mir sagen „So, jetzt war ich dran – nun darfst du tippen“. Nun gut, wenn ich soll? Ich tippe

LIST RETURN

und hurtig schreibt ER am Bildschirm

```
>
10 REM VERSION KOENIG
20 REM 84-11-12
30 CLS
40 PRINT " ..... * * * ....."
50 PRINT " * * * * * ."
60 PRINT " * * * * * ."
70 PRINT " * * * * * ."
80 PRINT " * * * * * ."
90 PRINT " ..... * * * ....."
100 PRINT
110 REM Begrueessung
115 REM -----
120 PRINT "Ich berechne Deinen taeglichen Umsatz"
130 PRINT "an zugefuehrter und verbrauchter Energie"
140 PRINT : PRINT "Willst Du weitermachen? (J/N)"
150 GOSUB 11000 : REM Antwort holen
160 IF TASTE$="N" THEN END
170 REM Definitionen
180 REM -----
190 ON ERROR GOTO 10000 : REM Fehlerbehandlung
200 CH$=CHR$(13)
210 SPEISEZAHL=100
220 DIM SPEISE$(SPEISEZAHL), ENERGIEZU(SPEISEZAHL)
230 ARBEIT=50
240 DIM ARBEIT$(ARBEIT), ENERGIE(ARBEIT)
250 MAHLZEIT=6
260 REM Zu welcher Arbeit wird die Speise gegessen?
270 DIM KATEGORIE(MAHLZEIT, SPEISEZAHL)
280 DIM MAHLZEIT$(MAHLZEIT)
290 MAHLZEIT$(1)="Fruehstueck"
300 MAHLZEIT$(2)="Jause"
310 MAHLZEIT$(3)="Mittagessen"
320 MAHLZEIT$(4)="Kaffeejause"
330 MAHLZEIT$(5)="Abendessen"
340 MAHLZEIT$(6)="Kleine Suende zwischendurch"
350 PRINT
360 INPUT "Wie ist dein Name"; ZEICHEN$
```

escape

&gt;

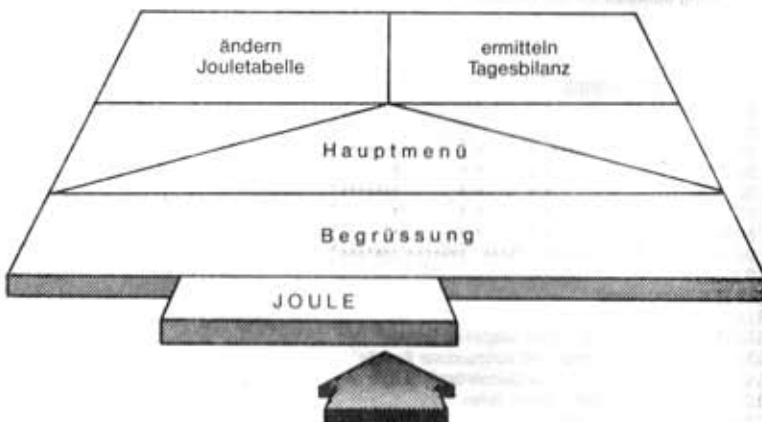
Schnell die ESCAPE-Taste gedrückt, denn sonst hört ER überhaupt nicht mehr auf. Nun sehen wir doch einmal nach, was ER da alles aufgeschrieben hat. Naja, ist doch gar nicht so schlimm. Mit der ersten Zeile

10 REM VERSION KOENIG

wird nur ein wenig erklärender Kommentar zu dem Programm angegeben. Und mit der Programmzeile 30

30 CLS

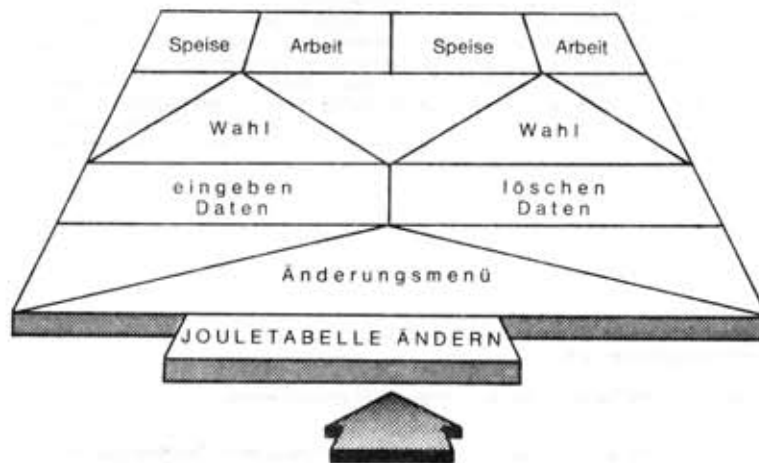
wird einfach der Bildschirm gelöscht, um die Arbeitsfläche unseres COMPUTERS freizumachen. Gar nicht so schwer, so ein Programm. Aber enorm viele Zeilen. Wer soll denn da noch den Überblick behalten. Zum Glück gibts STRUKTOGRAMME. Einfache Tafeln, auf denen der Ablauf des Programmes in groben Blöcken dargestellt ist. Nun wollen wir uns die erste Tafel betrachten.



Also das versteht doch jedes Kind. Zuerst begrüßt uns der COMPUTER schön artig und stellt uns sein Hauptmenü, eine Auswahl von Möglichkeiten, vor. Zum Beispiel kann mit der Entscheidung für die „Tagesbilanz“ ein regelrechter Dialog mit dem Computer begonnen werden. Dabei wird uns ein weiteres Menü vorgesetzt, das Sie aber zur Gänze konsumieren müssen, um Ihre Joulebilanz festzustellen.



Und wenn Sie dabei feststellen, daß Ihr Computer so manches Lieblingsessen von Ihnen noch nicht kennt, so kann Ihre kulinarische Spezialität einfach in die Speisenliste des COMPUTERS aufgenommen werden. Wie? Das zeigt das nächste STRUKTOGRAMM.



Sie wollen nun sicher wissen, wie wir ihm unser Schmäckerl mitteilen. Ganz einfach. Die Portion SCHINKEN, die wir so gerne zum Frühstück verspeisen, wird mit ihrem Namen und der Mengenangabe pro 100 g angegeben. Zusätzlich muß noch vermerkt werden, daß sie zum Frühstück gedrückt wird. Unser COMPUTER legt sich unsere Mitteilung in sogenannten DATA-Zeilen ab.

DATA 1, SCHINKEN, 335, 1, 0, 0, 0

Dabei bedeutet der erste Einser, daß es sich um eine Nahrung handelt. Die vier Zahlen zum Schluß der DATA-Anweisung kennzeichnen die Mahlzeit, zu der unser COMPUTER uns den Schinken anbietet. Der einzige Einser steht für „Frühstück und Kaffeejause“. Die drei weiteren Positionen sind für die Jause und das Abendessen, das Mittagessen und die „kleinen Sünden zwischendurch“, gedacht.

Unser COMPUTER ist aber auch fair und stellt unsere täglichen Arbeiten und körperlichen Anstrengungen den Genüssen gegenüber. Ob sie nun radfahren oder bügeln, alles wird mit Joule pro Stunde zu Buch geschlagen. Natürlich auch die Zeit des Ruhens! Also aufgepaßt, ER registriert jedes Nickerchen zwischendurch! Und berechnet's Ihnen mit 65 JOULE pro Stunde.

Falls Sie ein Floppy-Disk Laufwerk besitzen, dann speichert unser COMPUTER die aktuelle Kalorientabelle auf der Magnetplatte, sodaß jede Änderung unserer Speisen- und Tätigkeitstabelle ja nicht verlorengeht. Wie, Sie sind sich über die Begriffe „FLOPPY-DISK“ nicht im klaren? Macht nichts! Ist in Nullkommanix erklärt.

### Diskettenspeichersystem

Ihr Acorn Computer ist, wie auch andere Mikrocomputer mit einem internen Arbeitsspeicher ausgerüstet, der aus RAM's, Random Access Memories, aufgebaut ist. Ein in diesem Arbeitsspeicher abgelegtes Programm oder darin enthaltene Daten, können von

der Zentraleinheit, dem Mikroprozessor, bearbeitet werden. Sobald allerdings die Stromversorgung ihres Computers ausgeschaltet wird, geht alle Information im RAM verloren. Um Programme, aber auch Daten zu archivieren, muß der Computer in der Lage sein, den Inhalt des Arbeitsspeichers vor dem Ausschalten in eine Speichereinrichtung zu übertragen, die Informationen unabhängig von der Stromversorgung behalten kann.

An ihren Acorn B Mikrocomputer sowie an den Electron Computer, können als Externspeichermedium Kassettengeräte und Floppy-Disk angeschlossen werden. Auf beiden Speichermedien, Magnetband oder Diskette können sowohl Programme wie auch Daten direkt aus BASIC Programmen abgespeichert werden. Für beide Speichervorgänge, das Ablegen von Programmen sowie das von Daten, stehen unterschiedliche Befehle zur Verfügung. Während Programme auf Diskette und Band durch Funktionen des Betriebssystems abgespeichert und von dort wieder zurückgelesen werden, erfolgt die Übertragung von Zahlen und Zeichen aus Programmen auf das Externspeichermedium durch Anweisungen der Programmiersprache BASIC.

Die Speicherung auf Kassetten ist mit einer Reihe von Schwierigkeiten verbunden:

- Der Speicher- wie auch der Ladevorgang ist langsam.
- Der Bediener muß sich merken an welche Stelle auf dem Magnetband die Information geschrieben wird, da sonst die Gefahr besteht, daß Daten oder Programme irrtümlich überschrieben werden.
- Das Band muß händisch auf die richtige Stelle gespult werden.

Diese Probleme entfallen bei einem Diskettenspeichersystem. Der Bediener muß nur mehr die richtige Diskette, das sind flache biegsame Magnetscheiben in einer Schutzhülle, in das Diskettenlaufwerk einschieben. Alle Vorgänge im Laufwerk werden vom Computer mit Hilfe von Funktionen des Betriebssystems überwacht und gesteuert.

Das Abspeichern eines BASIC Programmes erfolgt beispielsweise mit dem Befehl SAVE. Dabei unternimmt das Betriebssystem für den Bediener folgende Schritte:

- Einschalten des Laufwerkes
- Auffinden eines freien Platzes von ausreichender Größe
- Eintragen dieses Speicherplatzes in das Inhaltsverzeichnis um das abgespeicherte Programm wiederfinden zu können
- Bewegung des Schreib-/Lesekopfes auf die zu beschreibenden Sektoren
- Übertragung der Programmkopie aus dem Arbeitsspeicher auf die Diskette
- Ausschalten des Laufwerkes

### Disketten

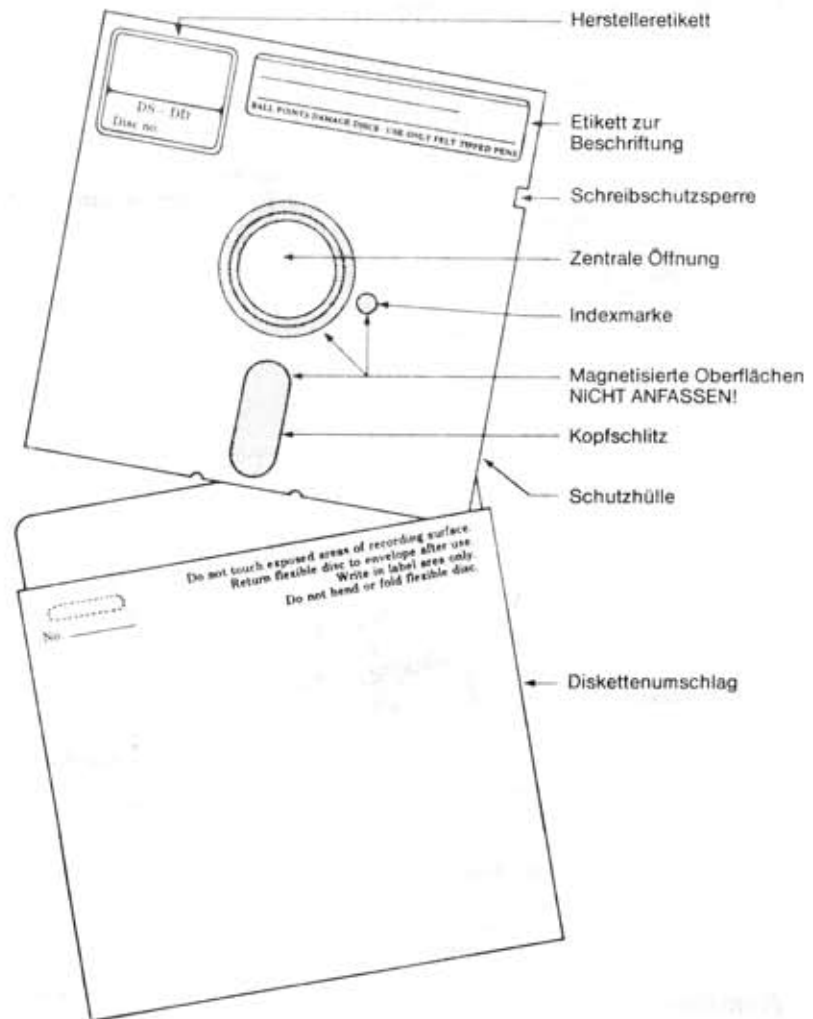
Der Acorn B Mikrocomputer verwendet zum Speichern von Informationen 5¼ Zoll Floppy-Disk, man bezeichnet sie auch als Mini Floppys.

Mit dem Acorn B Mikrocomputer können zwei Arten von Laufwerken eingesetzt werden. Das Einzellaufwerkssystem besteht aus nur einer Aufnahmevorrichtung für Disketten, mit diesem Laufwerktyp können 100 Kilobytes, das entspricht etwa 100.000 Zeichen, aufgezzeichnet werden.

Das Doppellaufwerkssystem besteht aus zwei unabhängigen Laufwerken und damit zwei Aufnahmeschlitzen für Disketten. Diese Laufwerkstypen kann auf einer Diskette 400.000 Zeichen speichern.

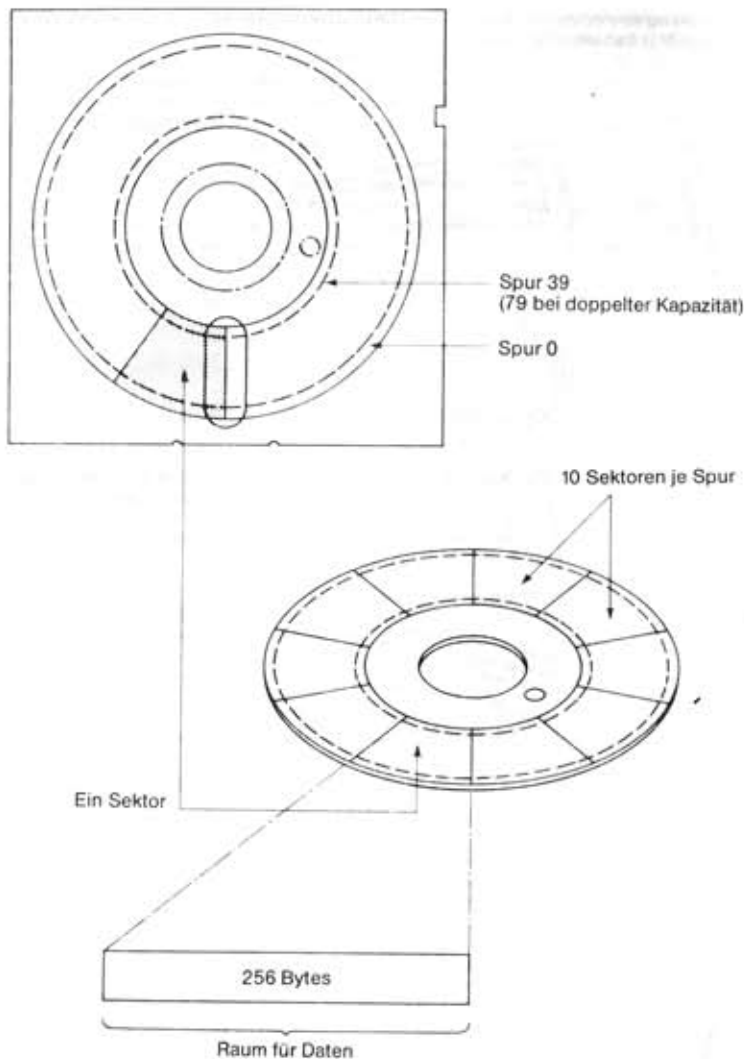
Disketten sind in der Regel sehr dicht mit Informationen beschrieben und reagieren aus

diesem Grund sehr empfindlich gegen kleinste Kratzer und Fremdkörper. Sie sollten daher sehr sorgfältig behandelt werden.



### Spuren, Sektoren und Bytes

Das Beschreiben von Disketten erfolgt in konzentrischen Kreisen, die auch als Spuren bezeichnet werden. Jede Spur ist beim Acorn B Mikrocomputer in 10 Sektoren aufgeteilt, diese Sektoren wiederum können je 256 Bytes aufnehmen. Die Speicherkapazität der Diskette wird wie die des Computers in Bytes angegeben. Ein Byte wird auch benötigt um ein darstellbares Zeichen des Alphabets abzuspeichern. Die 256 Bytes in einem Sektor stehen dem Anwender zum Abspeichern von Programmen oder Daten zur Verfügung.

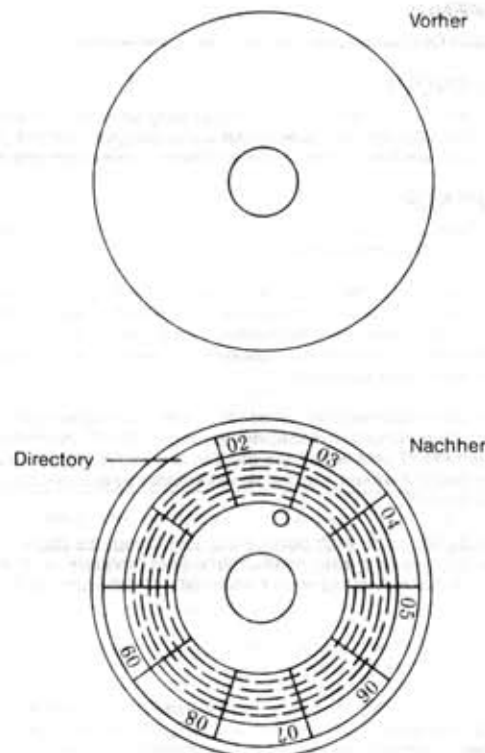


### Formatieren

Das Diskettenspeichersystem erstellt automatisch ein Inhaltsverzeichnis der auf der Diskette abgelegten Informationen. Dieses Inhaltsverzeichnis wird auf den beiden ersten Sektoren der Spur 0 abgespeichert die für diesen Zweck reserviert ist. Jedes Mal wenn eine bestimmte Datei von der Diskette gelesen werden soll, liest das Betriebssystem diese Sektoren in der Spur 0 und gewinnt daraus die Information, auf welchen Spuren und Sektoren sich die zu ladenden Informationen tatsächlich befinden.

Damit die aufgesuchten Spuren und Sektoren auch richtig identifiziert werden können, tragen sie selbst eine Markierung, die angibt um welchen Sektor, welche Spur es sich bei dem gegenständlichen Sektor handelt. Alle neuen Disketten müssen daher mit dieser Information beschrieben werden. Man nennt diesen Vorgang „formatieren“ und meint

damit das Markieren der Spuren und Sektoren auf der Diskette sowie das Abspeichern eines leeren Inhaltsverzeichnisses, des sogenannten Directory.



### Dateien

Unter Dateien versteht man Gruppen, bestehend aus Zahlen oder Wörtern, die zusammengefaßt werden. Ein einfaches Beispiel für eine Datei ist ein BASIC Programm, es ist eine zusammengehörige Gruppe von Anweisungen der Programmiersprache BASIC.

Programme werden durch die Betriebssystembefehle SAVE und LOAD zum und vom Externspeichermittel übertragen.

Zahlen und Zeichen können aber auch aus einem BASIC Programm in Dateien als eine zusammengehörige Gruppe von Informationen gestellt werden. Es kann sich dabei um Rechenergebnisse, Texte, Listen usw. handeln. Zur Verwaltung dieser aus BASIC zugreifbaren Dateien stehen eine Reihe von Anweisungen zur Verfügung. Diese Anweisungen können gleichermaßen für Kassetten und Diskettenversionen verwendet werden. Je nach Art des geladenen Betriebssystems werden die Daten dann auf Diskette oder Kassette übertragen. Daraus ergibt sich, daß für Kassettensysteme erstellte Programme im allgemeinen ebenso auf Diskettensystemen funktionieren.

Die Kommunikation zwischen BASIC und einer Datei erfolgt in drei Schritten:

- Eröffnen eines Datenkanals
- Datenübertragung
- Schließen des Datenkanals

Bevor BASIC auf eine Datei zugreifen kann, muß ein sogenannter Datenkanal eröffnet

werden. Dazu wird die BASIC Funktion OPEN verwendet. Diese Funktion steht im BASIC des Acorn B Mikrocomputers sowie des Electron Computers in drei unterschiedlichen Versionen zur Verfügung.

### Funktion OPENIN

Durch die Funktion OPENIN wird eine Datei zum Lesen eröffnet.

### Funktion OPENOUT

Eröffnen einer Datei zum Schreiben. Bei Ausführung der Funktion wird eine eventuell schon am Externspeichermedium bestehende Datei gleichen Namens gelöscht, existiert die Datei noch nicht, wird der Name in das Inhaltsverzeichnis der Diskette eingetragen.

### Funktion OPENUP

Eröffnen einer Datei zur Bearbeitung. Eine mit OPENUP eröffnete Datei kann sowohl gelesen als auch beschrieben werden.

Durch die drei Funktionen OPEN wird einem angegebenen Dateinamen eine sogenannte Kanalnummer zugeordnet. Wenn die Datei durch OPENOUT neu eröffnet wird, wird ihr Name im Directory des Externspeichermediums eingetragen. Die weiteren Operationen im BASIC Programm beziehen sich nur mehr auf die zugeordnete Kanalnummer, der Dateiname wird dann nicht mehr verwendet.

Die eigentliche Datenübertragung zwischen dem Externspeichermedium und dem Arbeitsspeicher erfolgt durch die BASIC Anweisungen INPUT und PRINT. Ähnlich wie bei den Anweisungen PRINT und INPUT, die auf den angeschlossenen Bildschirm bzw. die angeschlossene Tastatur wirken, werden Variableninhalte zwischen dem Arbeitsspeicher und Dateien transferiert.

Darüber hinaus stehen auch noch Befehle zur Verfügung, die nicht nur Zeichenketten zwischen Variablen und dem Externspeichermedium, sondern auch einzelne Zeichen einer Datei schreiben bzw. lesen können. Es handelt sich dabei um die Anweisungen BGET und BPUT.

Mit Hilfe der Funktion PTR wird in einer Datei die Stelle aufgesucht die dann durch PRINT direkt beschrieben oder INPUT gelesen wird.

Die BASIC Anweisung CLOSE dient zum Abschluß der Datenübertragung zwischen Datei und einem BASIC Programm. Durch die Anweisung CLOSE wird der Datenkanal freigegeben, die letzten Informationen werden auf das Externspeichermedium übertragen und die Datenübertragung beendet. Nach Ausführung der Anweisung CLOSE ist die Kanalnummer dem verwendeten Dateinamen nicht mehr zugeordnet. Erst eine neuerliche Zuordnung durch eine der möglichen OPEN Funktionen, ermöglicht neuerlichen Dateizugriff.

```

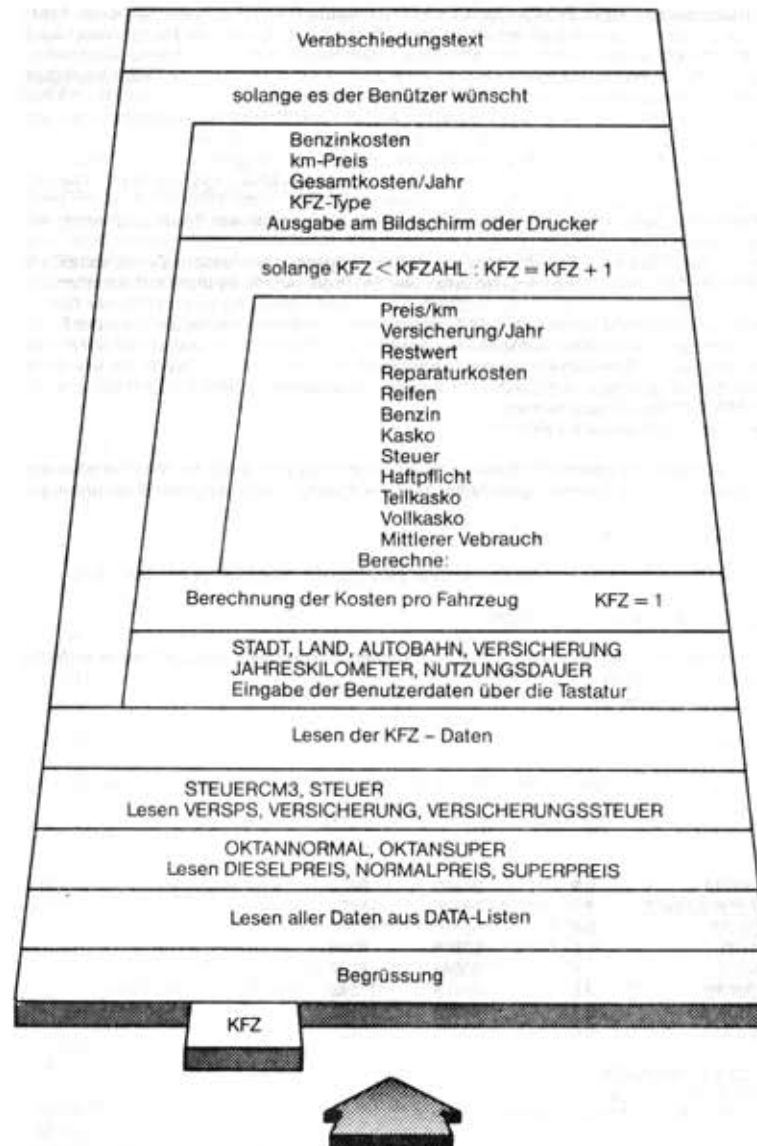
5 DIM Kino$ (3)
10 X = OPENIN ("Kinos")
20 B = 1
30 FOR A = 0 TO 20 STEP 10
40 PTR+ X = A
50 INPUT+ X, Kino$ (B)
60 B = B + 1
70 NEXT A

```

In einem BASIC Programm können bis zu fünf Dateien gleichzeitig eröffnet werden. Dazu müssen natürlich fünf unterschiedliche Kanalnummern verwendet werden.

## FOLGE 7 „NICHTS FUNKTIONIERT“

Sie kennen sicher das Gefühl, das uns packt, wenn wir in unser vielgeliebtes vierrädriges Etwas steigen und den Startschlüssel drehen – und nichts bewegt sich. Zuerst brandet es in uns auf – es wär doch gelacht, wenn unser „Eisenkamerad“ gerade heute verweigert. Nun gut, der kleine Rostschaden hinter dem linken Hinterrad, über den wollen wir nicht reden. Aber gar keine Reaktion, nicht mal bei gutem Zureden? Inzwischen ist nur mehr ein



jämmerliches Winseln zu hören und unser Herz rutscht schon beträchtlich tief nach unten. Soll es nun denn aus sein? Gib't nicht! Also raus aus dem Wagen, die Motorhaube auf und kräftig zugelangt. Zündkerzen – na ja – vielleicht der Verteiler! Aufgemacht und – halt, da ist auch schon der Finger des Verteilers irgendwo im Motor verschwunden. Wenn's bloß nicht so viele Winkeln in einem Auto gäbe. Auch egal, wir versuchen's nochmals! Also hinters Lenkrad geklemmt und der Schlüssel gedreht – nichts, überhaupt nichts. Da, jetzt fängt's auch noch zu regnen an! Hilft nichts, es heißt Abschied nehmen. Grad jetzt, wo ich

so knapp bei Kasse bin. Er hätte doch noch ein wenig durchhalten können! Ein neues Auto? Ja, ich wüßte schon eines, so mit Allrad und Schiebedach. Obwohl ein kleiner Flitzer auch nicht schlecht wäre. Doch mit den Koffern auf dem Bauch durch die Gegend rauschen ist auch nicht das Wahre! Vielleicht ein bequemer Kombi, mit dem ich meine Selbstbaumöbel praktisch transportieren könnte? Aber der Benzinverbrauch, bei den Preisen! Wenn's halt nicht so ein herrliches Vergnügen wäre, mit Metallic und Stereoradio dahinzuschweben. Doch jetzt ist Schluß mit dem Träumen. Diesmal wird ein Auto gekauft, das bei einem erträglichen Anschaffungspreis sparsam mit dem Benzin umgeht. Die KFZ-Steuer muß günstig sein und die Versicherung darf nicht zu hoch sein. Mein Gott, was man da bei der Auswahl eines Autos alles berücksichtigen muß. Sie - da könnte mir doch mein COMPUTER behilflich sein? Also raus aus dem bewegungslosen Etwas und nichts wie nach Hause.

Doch wie geh' ich's an? Es hilft nichts, ich muß ein Programm schneiden. Zuerst soll ER alle wichtigen Autodaten aus einer Liste lesen, wie den Preis, die PS, dann die Kubikzentimeter, den Benzinverbrauch, usw. Dann natürlich die Kosten für Versicherung und Steuer. Gut so. Weiters muß ich IHM sagen, wieviel Kilometer ich im Durchschnitt eines Jahres in der Stadt, auf dem Lande und auf der Autobahn herumkurve. Die Benzinpreise, die darf ich auf keinen Fall vergessen. Oder hab ich die schon erwähnt? Jetzt bin ich total verwirrt. Ich werde mir eine Skizze anlegen. Am besten in einem sogenannten STRUKTOGRAMM, wie die COMPUTER-Freaks das nennen. Am besten, ich versuch's einmal!

Puh, jetzt bin ich geschafft! Aber es müßte doch klappen, oder nicht? Nun aber ans Programmieren. In welcher Sprache? Natürlich in BASIC. Gleich die erste Zeile eingetippt



```
10 REM KFZ - Programm RETURN
```

So, der Programmtitel wär als Kommentar eingegeben. Sodann die nächste Zeile



```
20 CLS : REM Schirm löschen RETURN
```

mit der wir den Schirm löschen. So geht's munter dahin und schon ist das Programm fertig. Sie wollen's sehen? Sollen Sie haben. Mit



```
RUN RETURN
```

schreibt unser fleißiger COMPUTER am Schirm



KFZ - TYPE	KM-PREIS	KOSTEN/J	BENZIN
PRAERIE	2.3	28485	8980
MERCEDES 240D	3	36325	11734
SUBARU	2.6	31810	11561
R4-GTL	1.8	22596	8085
VW-POLO	1.6	20048	8087
PORSCHE	10.5	126116	17342
HONDA AC	2.3	28778	10181
CX-25-DI	3.6	43991	10427

NOCH EINE BERECHNUNG?  
HABEN SIE GUT GEWÄHLT?  
SOLLTEN SIE NICHT DOCH LIEBER SPAREN?

### Anweisung: DATA - Daten

#### Beschreibung:

Die BASIC Anweisung DATA ist ein Programmkennwort, das jeweils zu Anfang einer Datenliste stehen muß, damit diese mit der READ Anweisung gelesen werden kann.

DATA dient zur automatischen Bereitstellung von Eingabedaten - Zahlen oder Wörter - wenn diese innerhalb eines Programmes enthalten sein sollen.

Beispiel:

```
100 DATA "MEIER, HANS", 621, "JULIUS"
```

### READ - LESEANWEISUNG

Die Anweisung READ dient dazu, Elemente aus einer Datenliste in die mit dem Schlüsselwort READ verwendeten Variablen zu kopieren. Die Anweisung DATA muß zur Zuordnung der jeweiligen Zeichenketten und numerischen Variablen die Zeichenketten oder Zahlen in genau der Reihenfolge enthalten, in der sie in die Variablen übertragen werden sollen.

Die Anweisung READ liest die Werte, numerische oder Zeichenkettenausdrücke, die bei der Verarbeitung eines Programmes erforderlich sind, im Gegensatz zur INPUT Anweisung nicht von der Tastatur, sondern nimmt diese Werte aus DATA Anweisungen innerhalb des Programmes. Die READ Anweisung muß stets in Kombination mit DATA verwendet werden. Durch READ Anweisungen gelesene Elemente aus einer DATA Zeile, gelten im Augenblick der Verarbeitung als „erledigt“. Bei jedem nachfolgenden Lesen durch eine neue READ Anweisung, oder dieselbe READ Anweisung in eine Schleife gestellt, werden jeweils die nächsten Elemente aus der DATA Zeile gelesen. Wenn mit Hilfe einer oder mehrerer READ Anweisungen versucht wird in einer DATA Zeile mehr Elemente zu lesen als in dieser DATA Zeile enthalten sind, führt dies zur Ausgabe einer Fehlermeldung.

Beispiel:

```
100 READ A$, X
```

Die Anweisungen READ und DATA sollen anhand eines Beispiels erläutert werden. DATA Zeilen werden im Zusammenhang mit der READ Anweisung überall dort verwendet, wo es gilt feststehende Werte oder Zeichenketten, Variablen zuzuordnen. Ein Beispiel dafür ist die Zuordnung der Bezeichnungen der Kalendermonate

JANUAR, FEBRUAR...

zu ihren Monatszahlen

1, 2... bis 12.

Dazu ist es notwendig in einem Programm die Wörter Jänner bis Dezember abzulegen. Unser Programmbeispiel zeigt einen Programmteil, in dem über Tastatur eine Monatsziffer eingegeben werden kann und das entsprechende Wort aus DATA Zeilen ausgelesen wird. Dazu wird in einer FOR/NEXT Schleife die READ Anweisung der Zeile 50 so oft wiederholt, als durch die Monatsziffer angegeben ist. Dadurch kommt man beispielsweise für den Monat 6 durch sechsmaliges Lesen in der Zeile 50 zum sechsten Element der Datazeile, das den Text Juni enthält.

```
5 REPEAT
10 PRINT "GIB DEN MONAT ALS ZAHL EIN"
20 INPUT M
30 UNTIL M > 0 AND M < 13
40 FOR X = 1 TO M
50 READ A$
60 NEXT X
70 PRINT "DER MONAT IST": A$
100 DATA JANUAR, FEBRUAR, MAERZ, APRIL
110 DATA MAI, JUNI, JULI, AUGUST, SEPTEMBER
120 DATA OKTOBER, NOVEMBER, DEZEMBER
> RUN
GIB DEN MONAT ALS ZAHL EIN
? 6
DER MONAT IST JUNI
```



## RESTORE – Rückstellen des DATA Zeigers

### Beschreibung

Die Anweisung RESTORE dient dazu, den Datenzeiger einer DATA Zeile an den Anfang zurückzustellen.

Wenn die BASIC Anweisung RESTORE allein, ohne weitere Angabe verwendet wird, wird der Datenzeiger der jeweils ersten DATA Zeile eines Programms an den Anfang zurückgestellt. Unabhängig davon wieviele Elemente dieser DATA Zeile im Programmablauf durch READ Anweisungen schon gelesen wurden, führt dies dazu, daß bei Auftreten der nächsten READ Anweisung wieder mit dem ersten Element begonnen wird.

Wenn neben RESTORE eine Zeilennummer angegeben wird, führt dies dazu, daß der Datenzeiger auf den Beginn der DATA Zeile in der angegebenen Zeilennummer versetzt wird.

### Beispiel:

```
200 RESTORE 50
```

## Formatieren und steuern des Positionsanzeigers – Anweisung PRINT.

Zur Ausgabe von Zeichen auf Bildschirm oder Drucker wird die Anweisung PRINT verwendet. Die nach der Anweisung PRINT in Form einer Variablenliste oder in Form von Strings enthaltenen Werte werden nebeneinander am Bildschirm oder am Drucker ausgegeben. Zur Ausgabe von formatierten Listen ist es notwendig, die Position bestimmen zu können, an der ein bestimmtes Element der Druckliste ausgegeben wird. Eine einfache Formatierung wird schon durch die Wahl der Trennzeichen der Elemente in der Druckliste durchgeführt.

Der Bildschirm ist in Spalten eingeteilt, die jeweils eine Breite von 10 Zeichen haben. Das bedeutet, die erste Spalte beginnt an der ersten Position, die zweite an der elften, die dritte an der einundzwanzigsten, usw. Wenn Elemente einer Druckliste durch Beistriche getrennt sind, wird die Ausgabe des jeweils nachfolgenden Elementes in der nächsten Spalte durchgeführt.

```
10 X = 8
20 A$ = "HALLO"
30 PRINT X, X, X
```

```
> RUN
```

```
8      8      8
```

Unser Beispiel zeigt die dreifache Ausgabe der Variablen X, die die Zahl 8 enthält, jeweils in die zehnte, zwanzigste und dreißigste Position. Werden als Trennzeichen Strichpunkte verwendet, werden die Zahlen direkt nebeneinander geschrieben.

Wenn die Formatierung der Druckliste durch Verwendung von Beistrichen erfolgt, werden Zahlen rechtsbündig in die Spalten eingetragen, Wörter dagegen linksbündig.

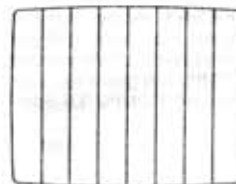
```
10 X = 8
20 A$ = "HALLO"
30 PRINT X, X/2, X/4
40 PRINT A$, A$, A$
```

```
> RUN
```

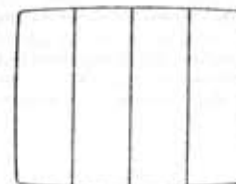
```
HALLO      HALLO      HALLO
           |           |
           |Feldbreite|
```

Das Wort HALLO ist jeweils am Beginn einer Spalte, die Zahlen 8, 4, und 2 am rechten Ende einer Spalte angeordnet.

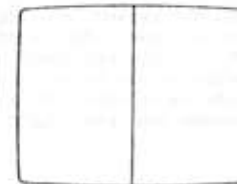
Die Spaltenbreite ist automatisch auf zehn Zeichen festgesetzt. Da im Acorn 8 Mikrocomputer die Gesamtspaltenbreite am Bildschirm je nach Betriebsart (Modus) zwischen 80 Zeichen und 20 Zeichen variiert, ergibt sich daraus eine unterschiedliche Spaltenanzahl pro Betriebsart.



80-Zeichen-Modi  
(Modi 0 und 3)



40-Zeichen-Modi  
(Modi 1, 4, 6 und 7)



20-Zeichen-Modi  
(Modi 2 und 5)

## Funktion TAB

Die Gestaltung der Wiedergabe auf Bildschirm und Drucker läßt sich nicht nur durch die Trennzeichen Beistrich und Strichpunkt bestimmen. Die Funktion TAB kann in einer Druckliste dazu verwendet werden, um die Ausgabe an einer bestimmten Spaltenposition fortsetzen zu können. Wenn die Funktion TAB in einer PRINT Anweisung vorkommt, wird der Cursor an die angegebene Position gesetzt und dort mit der Ausgabe fortgesetzt.

```
10 PRINT "012345678901234567890"
20 F = 16
30 REPEAT
40 PRINT TAB (10); F; TAB (15); 2 * F
50 F = F + 1
60 UNTIL F = 18
>
> RUN
012345678901234567890
          16 32
          17 34
```

## „MEINE SCHALLPLATTENSAMMLUNG“

## FOLGE 8

Also gestern hat es mir gereicht. Den ganzen Tag im Streß, jeder wollte etwas von mir. Abends müde heimgekehrt war mein erster Wunsch – totale Entspannung, zu mir finden und wieder Mensch werden. Sie können sich's vorstellen, ja? Haben Sie's schon einmal mit Sibelius oder Tschaiakowsky versucht? Herrlich, sage ich Ihnen. Also nichts wie hin zum Plattenschrank und nach der „Seelen-Medizin“ gegriffen. Was halte ich da in meiner Hand – die „Stones“! Nur die jetzt nicht. Nach einer Zeit heftigen Stöberns sinke ich frustriert in den Sessel. Doch da kommt der Überlebenswille hoch. Da muß einfach einmal Ordnung in meine Schallplatten-Sammlung gebracht werden. Aber wie? Soll ich nach den Interpreten ordnen? Oder nach der Muskart, Pop und Klassik getrennt? Das merk' ich mir nie! Ich habe eine Idee. Ich nummeriere die Platten, so wie sie in der Ablage stehen, der Reihe nach durch. Dann kann ich sie leicht aufstöbern und auch wieder einordnen. Und wenn ich eine Platte geschenkt bekomme, dann wird sie einfach drangereiht und bekommt eine neue Nummer. Gut, aber die Nummer sagt noch nichts über die Platte selbst aus. Also muß eine Liste angelegt werden, wo zu jeder Nummer alle weiteren Angaben der einzelnen Platten angegeben werden. Hui, das wird bei meiner Sammlung ganz schön aufwendig und vor allem mühsam, die richtige herauszufinden. Mein müder Blick fällt auf meinen

COMPUTER. Zwinkert er mir nicht mit seinem matten Bildschirm und der glänzenden Tastatur zu? Na klar, die Plattenkartei soll doch ER für mich verwalten. Na servus, das wird ein Monsterprogramm. Da seh' ich mich schon vor dem Bildschirm und die Morgendämmerung steigt langsam hoch. Meinetwegen, nur heute nicht. Und trotzdem läßt's mich nicht in Ruhe! Da hab' ich doch irgendwo ein fix und fertiges Programm, mit dem so eine Sammlung spielend angelegt werden kann. Wo hab' ich dieses nur – da ist es. „DATABASE“ steht drauf und soll laut Aufschrift eine „elektronische Kartei“ sein. Okay, dann werde ich für jede Platte ein elektronisches „Karteiblatt“ anlegen, auf dem ich dann den Interpreten, den Komponisten, den Titel, das Produktionsdatum und natürlich meine Nummer eintragen werde. Wenn ich die Sache so betrachte, dann gehören doch auch noch die Musikart und die einzelnen Titeln drauf. Na, ob das mein Guter schaffen wird? Wir werden sehen. Jetzt nichts wie die Diskette ins Laufwerk und den COMPUTER gestartet. Laut Anleitung soll ER sich mit einem „Menü“ melden – Sie wissen schon, der COMPUTER-Speisezettel. Und da ist er schon

## DATABASE MAIN MENU

- 1 Create a new file
- 2 Examine/Update an existing file
- 3 Add a field to a file
- 4 Change field sizes/names
- 5 Convert a file to text
- 6 Make a View macro file

Your selection -

Natürlich können Sie diese Speisekarte hinauf und hinunter essen, aber schön langsam der Reihe nach. Wie, Sie haben Schwierigkeiten beim Lesen? Ach so, ich versteh, das Englisch. Gut, ich werde Ihnen ein wenig behilflich sein. Also, wenn sie den 1. Gang wählen, dann legen Sie ein neues Karteisystem an. Mit dem 2. Gang können Sie dann in Ihrer Kartei munter blättern und gegebenenfalls was dazu tippen. Mit dem 3. Gang kann das Karteisystem erweitert werden, mit dem 4. Gang werden Sie Ihre Kartei, falls notwendig, umstrukturieren und mit den letzten beiden Gängen kann Ihre Kartei, oder sagen wir's gleich fachlich richtig – DATEI, in eine Textform umgewandelt werden sowie als Basis für eine Adressen-Datei dienen. So, jetzt entscheiden wir uns gleich für die Nummer 1, damit unsere Kartei, DATEI natürlich, endlich entsteht. Hurlig eine 1 eingetippt und der Schirm meldet

## Initialise File

How many fields will  
each record need?  
(Maximum 32)

Die Frage lautet, wieviele Felder sollen für die Kartei reserviert werden – ich würde sagen 9. Getippt fordert der Schirm die Angabe der ersten Feldbezeichnung. Da muß ich sparsam sein, denn es sind 7 Buchstaben zugelassen. Ich beginne mit „INTERPR“ und gebe eine Feldlänge von 27 Buchstaben an. Ja, manche Künstler haben die ausgefallensten Namen. Nun fragt mich der Schirm nach der 2. Feldbezeichnung und so weiter. Nach langem überlegen hab' ichs drinnen. Da fordert mich mein COMPUTER auf, eine Datendiskette einzulegen, damit ER die Kartei, besser DATEI, für mich anlegen darf. Getan, das Laufwerk schnurrt leise, und schon zeigt sich wieder das Hauptmenü. So, jetzt werden wir das erste Karteiblatt beschreiben. Also wähle ich die Nummer 2 und der Bildschirm zeigt.

## Platten Record 1 Page 1/2

Interpr	
Kompnst	
Titel	
Nummer	
Kategor	
Datum	
Bemerkg	
Seite 1	
Seite 2	

Toll, was? Gleich die erste Platte aus dem Regal gefaßt und munter Dateistruktur eingetippt. Doch halt, zuerst kriegt sie noch eine Nummer. Warum nicht die Nummer 1? Getan – ein Blick auf den Interpreten – herrlich, ein Gospel mit den JONNY THOMPSON SINGERS. Ich bin ein Fan von Gospel- und Soulmusik, Sie auch? Gut, was kommt als nächstes? Ah ja, der Komponist – natürlich von Reverend Jonny Thompson. Das ist leicht eingetippt. So, jetzt noch der Titel. Da kann man ja gar nicht aufhören. In nullkommanix sind alle Daten vom Cover drin und das sieht dann so aus

## Platten Record 1

Interpr	The Jonny Thompson Singers
Kompnst	Jonny Thompson
Titel	Lord I Belong To You
Nummer	1
Kategor	Gospel
Datum	82-10-00
Bemerkg	

Seite 1

Lily Of The Valley, A Personal Testimony  
 Little Talk With Jesus, Mirror on The Wall  
 Lord I Belong To You, I Want To Be Ready

Seite 2

I'll Fly Away, Look At Me Now,  
 Mary and Martha, Come Here Lord  
 The Cross, Oh The Joy, Hold On - Hang On

Nicht schlecht, was? Sie wollen schon mitmachen? Na, dann erstellen Sie doch Ihre eigene Datei! Viel Erfolg!

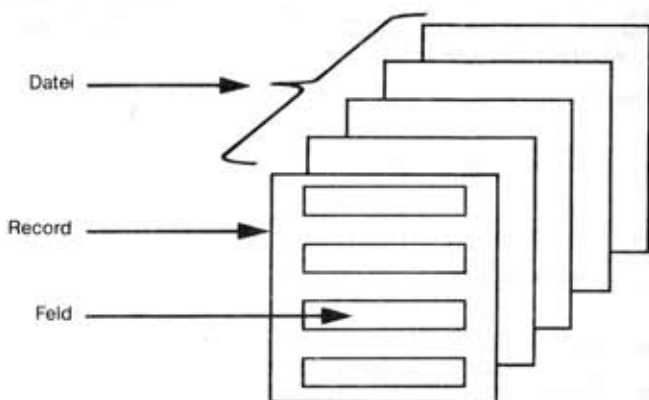
## Datenbanksysteme

Die Verwaltung von Daten in Dateien ist eine Aufgabe für die sich Computer besonders gut eignen. Vor allem im Zusammenhang mit Floppy Disk als Externspeichermedium können durch Mikrocomputer sehr effiziente Datenbanken realisiert werden.

Die auf Heimcomputern eingesetzten Programmiersprachen enthalten Anweisungen zum Übertragen von Informationen zwischen dem Arbeitsspeicher des Mikrocomputers und dem Externspeichermedium. Mit diesen Funktionen können Dateien eröffnet oder gelöscht werden sowie Datensätze in diese Dateien übertragen werden.

Bei der Verwaltung von größeren Informationsmengen treten neben der einfachen Aufgabe der Speicherung und des Rücklesens von Informationen zusätzliche Anforderungen an ein Computersystem auf.

Dateien werden dabei in sogenannte Records unterteilt. Records stellen eine Gruppe von zusammengehörigen Informationen innerhalb der Dateien dar. Jedes Record wiederum besteht aus Feldern, die mehrere Einzelzeichen, Zahlen oder Buchstaben aufnehmen können.



Die Records eines Files müssen von identischer Struktur sein. Das bedeutet, sie müssen aus derselben Anzahl von Feldern bestehen, die Felder müssen die gleiche Bezeichnung und gleiche Länge haben. Unser Beispiel zeigt eine Datei, die aus Karteikarten aufgebaut

ist. In einer Adreßdatei wäre ein einzelnes Record ein Blatt auf dem alle zu einer Person gehörenden Informationen enthalten sind. Die Informationen innerhalb des Records, die Felder, sind in unserem Beispiel der Name der Person, das Alter, die Telefonnummer, die Anschrift, usw.

Bei der Verwaltung von mehreren solcher Records treten auch bei unterschiedlichen Anwendungen gleiche Verwaltungsvorgänge auf. Records sollen beschrieben und in die Datei eingeordnet werden können. Records sollen aber auch in der Datei gesucht, angezeigt, ausgedruckt und auch verändert werden können. Häufig sollen die Records einer Datei auch nach einem bestimmten Feldinhalt sortiert werden. In unserem Beispiel einer Namensdatei z. B. nach der alphabetischen Reihenfolge der Namensfelder.

Zur Realisierung der erwähnten Funktionen sind komplexe Programmsysteme notwendig. Diese Programme sind, wenn sie mit Hilfe der einfachen dateibezogenen Anweisungen, die höhere Programmiersprachen beinhalten, geschrieben sind, sehr komplex. Es ist daher nicht sinnvoll für jeden Anwendungsfall einer Dateiverwaltung die Manipulation der zu verwaltenden Felder und Records durch ein selbst erstelltes Programm in höherer Programmiersprache zu realisieren. Diese Aufgabe wird durch das Programmwerkzeug DATENBANK bewerkstelligt.

Datenbanksysteme sind Programme, die die Verwaltung von Informationen in Dateien durchführen. Damit sie vom Bediener für unterschiedliche Anwendungen eingesetzt werden können, enthalten sie neben den Befehlen zur Eingabe und zur Manipulation von Informationen auch Befehle mit deren Hilfe die Datenbank an die Struktur der zu verwaltenden Informationen angepaßt werden kann.

### Es gibt zwei Arten von Datenbanksystemen.

Datenbankbibliotheken sind meist im Assembler Code eines Prozessors geschriebene Unterprogrammbibliotheken, die die wichtigsten Datenbankfunktionen zur Verfügung stellen. Diese Bibliothekselemente können jedoch vom Benutzer eines Computers nicht direkt aufgerufen werden, sie benötigen ein überlagertes Programm in höherer Programmiersprache. Dadurch hat der Anwender die Möglichkeit, ein spezifisches Verwaltungsprogramm zu schreiben, wobei die schwierigen Dateimanipulationsfunktionen nicht selbst programmiert werden müssen, sondern nur aus Aufrufen von Bibliotheksfunktionen bestehen.

Selbstständige Datenbanksysteme hingegen sind Programme, die über Tastatur und Bildschirm direkt vom Anwender bedient werden können. Sie benötigen keine weitere Programmumgebung im Computer. Nach dem Aufruf derartiger Programme erscheint am Bildschirm ein Funktionsmenü, das neben den Anpassungsfunktionen auch die Manipulationsfunktionen beinhaltet.

Auf dem Acorn B Mikrocomputer steht das Datenbanksystem „DATABASE“ zur Verfügung. Dieses Softwarewerkzeug muß zusammen mit einem Floppy Disk Speichersystem verwendet werden. Das Programm selbst ist auf einer Diskette gespeichert. Nach Aufruf des Programmes erscheint die Funktionsauswahl am Bildschirm.

```

Database Main Menu
1 : Create a new file
2 : Examine/Update an existing file
3 : Add a field to a file
4 : Change field sizes/names
5 : Convert a file to text
6 : Make a View macro file
Your selection _
  
```

Bevor Daten eingegeben werden, muß die Struktur der zu verwaltenden Daten bestimmt

werden. Dabei wird festgelegt, welche Informationen ein Record der Datei tatsächlich aufnehmen soll. In unserem Beispiel einer Namensdatei muß bestimmt werden, welche Informationen pro Person auf das Karteiblatt geschrieben werden soll und welche maximale Länge für jedes Feld vorgesehen ist.

Für eine Namensdatei könnte die Einteilung wie in unserem Beispiel aussehen:

Field number	Field name	Field length
1	SURNAME	20
2	NAME	20
3	ADDRESS	84
4	TEL NO	12
5	BTHDAY	6
6	CARD?	2

Jedes Record in „DATABASE“ kann maximal 32 Felder aufnehmen. Die maximale Anzahl von Zeichen in jedem Feld ist 240.

Die Festlegung der Recordstruktur einer Datei stellt eine wichtige Entscheidung für den Anwender einer Datenbank dar. Es gilt dabei die wichtigen Informationen auszuwählen und sicherzustellen, daß die entsprechenden Felder genügend Platz zur Aufnahme dieser Informationen bieten werden. Es muß aber auch darauf geachtet werden, daß für jedes Feld die kleinstmögliche Zeichenanzahl vorgesehen wird. Jedes Feld, sowie jedes Zeichen in jedem Feld, verbraucht, egal ob es beschrieben wird oder nicht, Platz in der Datei. Neben der Tatsache, daß durch großzügige Vergabe von Feldern und Zeichen in Feldern der verfügbare Speicherplatz pro Diskette schlecht ausgenützt wird, steht auch das Problem der Reaktionszeit des Datenbanksystems. Je größer die Informationsmengen sind, die ein Datenbanksystem in einem Such- oder Sortiervorgang bearbeiten muß, desto länger muß der Bediener natürlich darauf warten. Aus diesen Grundsätzen ergibt sich, daß der Anwender einer Datenbank sehr genau auswählen muß, welche Informationen pro Record gespeichert werden.

Nachdem der Anwender einer Datenbank die Recorderteilung getroffen hat, kann er diese Einteilung auf Diskette eintragen. Dazu wird in „DATABASE“ die Funktion

1 : CREATE A NEW FILE

aufgerufen.

Nach Aufruf dieser Funktion muß dem Computer die Gesamtanzahl der Felder pro Record mitgeteilt werden. Am Bildschirm erscheinen dann Informationszeilen für jedes Feld, wobei für jedes dieser Felder eine Bezeichnung, sowie die maximale Zeichenanzahl, die Feldlänge eingegeben werden kann. Nach der Definition aller Felder eines Records kann ein Name für die festgelegte Datei eingegeben werden. Unter diesem Namen wird die Recordstruktur der Datei auf dem Externspeichermmedium abgelegt. Nach Durchführung dieser Funktion befindet sich eine leere Datei mit der getroffenen Einteilung auf der Diskette.

Zur praktischen Arbeit mit Dateien ist es sinnvoll, für unterschiedliche Dateien auch unterschiedliche Disketten zu verwenden. In einem Acorn B Mikrocomputersystem mit zwei Floppy Disk Laufwerken muß die Programmdiskette immer im Laufwerk 0 sein. Es ist dann sinnvoll, die Datendiskette im Laufwerk 1 oder in einem System mit doppelseitig beschreibbaren Disketten im Laufwerk 3 zu verwenden.

Die Befehle von „DATABASE“ zur Eingabe und zum Ausdruck von Informationen aus dem Datenbanksystem beschreiben wir in den Unterlagen zur Folge 10 der Fernseh-Computerfamilie.

## FOLGE 9 „FUSSBALLFIEBER“

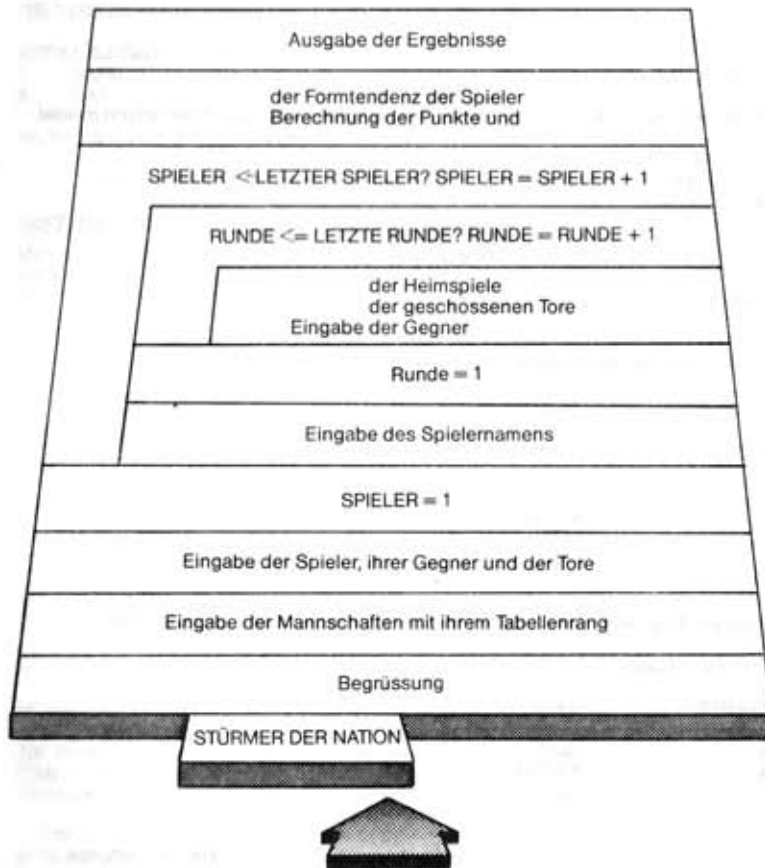
Stellen Sie sich vor, letzte Woche treffe ich einen guten alten Bekannten. Wie man so ins Reden kommt, erzählt er mir von seinem Fußballverein, den er treu zu jedem Spiel begleitet. Für mich ist sein Verein mittelklassig und das hab' ich ihm auch nicht verhehlt. Na, ein Wort gibt das andere – wir sind uns glatt in die Haare geraten. Denn ich bin der Meinung,

daß der Mittelstürmer seines Vereins um Klassen schwächer spielt als mein Favorit. Und da es um den Posten des Mittelstürmers in der Nationalmannschaft für das kommende Ländermatch geht, hat natürlich keiner von uns beiden nachgegeben. Da kam mir die Idee mit dem COMPUTER. Ich hab meinen Freund eingeladen, gemeinsam ein Bewertungsprogramm für die Auswahl des richtigen Stürmers zu erstellen. Soll doch der COMPUTER auch einmal heißlaufen, wenn es um so ein wichtiges Thema geht. Mein Freund willigte ein, betonte aber, daß, wie immer auch das Ergebnis der „Fußballerwahl“ ausfällt, es nachher heißen wird: ja, aber!

Zuhause angekommen machten wir es uns gemütlich und versuchten ein Punktesystem auszutüfteln. Anfangs war es recht schwierig, aber bald schon nahm die Idee konkrete Formen an.

Es wird die Meisterschaftstabelle eingegeben und ermittelt, wieviele Tore jeweils die Kandidaten in der Meisterschaft geschossen haben. Dieses Auswahlprinzip funktioniert natürlich nur für Stürmer. Bei der Berechnung wird berücksichtigt, ob die vom Stürmer geschossenen Tore gegen eine schwächere oder stärkere Mannschaft entsprechend der Meisterschaftstabelle erzielt wurden. Die Berücksichtigung erfolgt in Form eines Bonus/ Malus Punktesystem. Weiters wird bei der Bewertung ein Bonus vergeben, wenn die Tore in einem Auswärtsspiel geschossen wurden. Zu guter Letzt soll unser COMPUTER noch bewerten, ob sich ein Spieler in den letzten Spielen steigern konnte. Er erhält dafür ebenfalls einen Punktezuschlag.

Die Zeit verging wie im Flug, doch dann war es geschafft. Ein fertiges STRUKTOGRAMM lag vor uns auf dem Papier.



Auf den fragenden Blick meines Freundes in Richtung meines COMPUTERS mußte ich ihm klarmachen, daß nun erst die Programmierung des COMPUTER-Programms entsprechend dem STRUKTOGRAMM erfolgen mußte. Nun, ich will Sie nicht länger hinhalten, am selben Abend war unser Programm noch nicht fertig. Es dauerte schon ein paar Tage, bis die Fußball-Tabelle in DATA-Anweisungen verpackt war und die Programmschleifen anstandslos funktionierten. Dann war's soweit: Das Programm funktionierte! Mit dem Kommando



RUN RETURN

löschte der COMPUTER den Bildschirm und es erschien die erste Meldung



SUCHE NACH DEM BESTEN STUERMER DER NATION

DAS 1. MAL (J/N)



Klar ist's das erste Mal. Also tippen wir J für ja und schon murmelt das Diskettenlaufwerk. Nach kurzer Zeit erscheint



MANNSCHAFT            TABELLENRANG

RAPID	1
AUSTRIA	2
LASK	3
GAK	4



STIMMEN DIE PLATZZIFFERN? (J/N)

Wollen wir sagen, sie stimmen. Also tippen wir wieder J für ja und der Schirm meldet



BITTE JETZT EINGABE  
DER SPIELER  
IHRER GEGNER  
SOWIE IHRE TORE

WIEVIELE SPIELER SOLLEN VERGlichen WERDEN? 2  
WIEVIELE RUNDEN SOLLEN VERGlichen WERDEN? 2

NAME DES            1. SPIELER: ?KRANKL



Versuchen wir's mit dem „Bomber der Nation“, unserem Hans KRANKL? Gut, wir tippen KRANKL mit der Abschlußtaste RETURN und der Schirm meldet



IN WELCHER MANNSCHAFT SPIELT KRANKL

NUMMER            MANNSCHAFT

1	RAPID
2	AUSTRIA
3	LASK
4	GAK



Also, das weiß jedes Kind – bei RAPID natürlich! Rasch eine 1 getippt und es zeigt sich



GEGNER : 1 RUNDE:

NUMMER            MANNSCHAFT

1	RAPID
2	AUSTRIA
3	LASK
4	GAK

Das letzte Spiel, tja wer da der Gegner war? Jetzt weiß ich's – die AUSTRIA natürlich. Eitrig

getippt und der Schirm zeigt

```
?2
WIEVIELE TORE HAT KRANKL GEGEN AUSTRIA GESCHOSSEN ?3
HEIMSPIEL .....0
AUSSWAERTSSPIEL...1
?1
```

Ich glaub', es waren 2 Tore. Gut, eine 2 getippt und auswärts natürlich!

So wird in einem fort fleißig eingetippt. Schließlich sind die letzten Daten drinnen und unser COMPUTER schreitet zur Endabrechnung

PUNKTVERGLEICH

KRANKL VON RAPID HAT IN DEN LETZTEN 2 RUNDEN 4 TORE GESCHOSSEN  
UND DABEI 3.21142857 PUNKTE BEI FALLENDER FORM ERREICHT

RUNDE/GEGNER/PLATZDIFFERENZ/HEIM-AUS/TORE/PUNKTE
1 /AUSTRIA/ -1 / A / 3 / 2.04
2 /GAK / -3 / H / 1 / 0.571428571

PROHASKA VON AUSTRIA HAT IN DEN LETZTEN 2 RUNDEN 3 TORE GESCHOSSEN  
UND DABEI 2.68333333 PUNKTE BEI FALLENDER FORM ERREICHT

RUNDE/GEGNER/PLATZDIFFERENZ/HEIM-AUS/TORE/PUNKTE
1 /RAPID / 1 / H / 1 / 1.25
2 /GAK / -2 / H / 2 / 1.33333333

Na, hab' ich recht gehabt? Das wär's, Freunde!

## GET, INKEY – Einlesen von Einzelzeichen

Manche Anwendungen von Computerprogrammen erfordern die sofortige Reaktion des Computers, nachdem eine Taste gedrückt wird. Wenn die BASIC Anweisung INPUT zum Einlesen von Zeichen von der Tastatur verwendet wird, muß die Eingabe mit Return abgeschlossen werden, bevor das Programm mit der nächsten Anweisung fortsetzt. Vor allem bei Spielen, aber auch bei Auswahlentscheidungen bei der Bedienungsführung durch die sogenannte Menütechnik, muß eine Reaktion des Computers auch auf einen Einzeltastendruck ermöglicht werden. Dazu gibt es die Funktionen:

GET, GET\$  
INKEY, INKEY\$

Die Funktionen sind einander ähnlich und dienen dazu, bei Ausführung der Funktion die Tastatur abzufragen und den ASCII Code einer eventuell gedrückten Taste, an eine Variable im Programm zu übergeben. Je nachdem, ob das Drücken einer Ziffer, die als numerische Variable weiterverarbeitet werden soll, oder das Drücken eines Zeichens erwartet wird, wird GET bzw. GET\$ und INKEY bzw. INKEY\$ verwendet.

Die Funktionen GET und GET\$ arbeiten ähnlich wie die Anweisung INPUT. Wenn sie ausgeführt werden, wird die Programmausführung angehalten und ein Tastendruck abgewartet. Das Programm läuft allerdings nach dem Drücken der ersten Taste, deren Code an die zugeordnete Variable übergeben wird, mit der nächstfolgenden Anweisung weiter.

Mit den Funktionen INKEY und INKEY\$ kann ein Parameter bestimmt werden, der eine Frist enthält, während der das Programm bei Ausführung der Funktionen angehalten werden soll. Wenn innerhalb dieser Zeit eine Taste gedrückt wird, wird der entsprechende ASCII Code an die Variable übergeben. Wird innerhalb dieser Zeit keine Taste gedrückt, geht der Computer zur nächsten Programmzeile über und die Variable bleibt leer.

```
1000 PRINT "WILLST DU WEITERMACHEN (J/N)"
1100 PRINT "DU HAST 2 SEKUNDEN ZEIT"
```

```

120 A$ = INKEY$ (200)
130 IF A$ = " " THEN PRINT "ZU SPAET -
VERPASST"
140 IF A$ = "J" THEN PRINT "WAGHALSIGER
DUMMKOPF"
150 IF A$ = "N" THEN PRINT "FEIGLING"

```

Unser Beispiel zeigt einen Programmausschnitt, in dem das Programm zwei Sekunden auf eine Reaktion wartet.

### Unterprogramme – GOSUB

Die Anweisung GOSUB erlaubt es, ähnlich wie die Anweisung GOTO, von einer beliebigen Programmzeile zu einer anderen zu springen. Im Gegensatz zur Anweisung GOTO wird bei Ausführung dieses Sprungbefehls jedoch die Zeilennummer abgespeichert, die ohne Ausführung des Sprunges im normalen Programmablauf als nächste ausgeführt worden wäre. Wenn der Programmablauf auf die BASIC Anweisung RETURN trifft, wird die letzte mit GOSUB gespeicherte Adresse, sie wird auch Rückkehradresse genannt, geladen und das Programm kehrt dorthin zurück, woher die Verzweigung durch GOSUB erfolgt ist.

In vielen Fällen muß ein Block von Zeilen innerhalb eines Programmes mehrere Male verarbeitet werden. Anstelle diesen Anweisungsblock mehrere Male an den entsprechenden Stellen einzugeben, kann man ihn als Unterprogramm definieren, das dann bei Bedarf innerhalb des Hauptprogramms durch die Verwendung der Anweisung GOSUB aufgerufen wird. Das Ende eines Unterprogrammes muß durch die Anweisung RETURN gekennzeichnet sein.

Ein Unterprogramm kann wiederum ein anderes aufrufen, auch sich selbst. Die Verschachtelung von Unterprogrammen darf im BASIC des Acorn B Mikrocomputers und im Electron 26 nicht überschreiten.

```

10 INPUT A
20 GOSUB 500
30 ...
.
.
115 A = X - 4
120 GOSUB 500
130 ...
.
.
500 B = (A/2 - SIN (A + 3))/(25 * A)
510 PRINT "DER J-WERT IST: ", B
520 RETURN

```

Unser Beispiel zeigt die Anwendung der Anweisung GOSUB in einem Programmausschnitt. Dabei wird an zwei verschiedenen Stellen eines Programmes eine Formel berechnet und ein Wert ausgegeben. Diese Programmzeilen sind als Unterprogramm zusammengefaßt.

Durch die Verwendung von Unterprogrammen läßt sich ein BASIC Programm vereinfachen und zusätzlich übersichtlich gestalten.

### Funktionen

Neben den Anweisungen sind die Funktionen ein wichtiges Element jeder Programmiersprache. Funktionen sind wie Anweisungen durch reservierte BASIC Schlüsselwörter gekennzeichnet.

Während eine Anweisung beliebige Vorgänge im Computer einleiten kann, je nach Art der gewählten Anweisung, führen Funktionen immer zur Berechnung eines Ergebnisses. Durch eine Funktion wird ein Ergebnis zur weiteren Verarbeitung wie z. B. Übertragung in

eine Variable oder Ausgabe bereitgestellt. Dieses Ergebnis kann sowohl eine Zahl, wie auch eine Zeichenkette sein. BASIC enthält bereits eine Anzahl von Funktionen, sie sind durch Schlüsselwörter gekennzeichnet. Die Funktion SQR berechnet beispielsweise die Quadratwurzel einer Zahl.

```

10 Y = SQR (16)
20 PRINT Y

10 PRINT SQR (Y)

```

Die beiden Teile unseres Beispiels führen die selbe Berechnung durch. Es wird die Quadratwurzel der Zahl 16, sie ist 4, berechnet und ausgegeben. Durch die Anweisung DEF können Funktionen vom Anwender selbst definiert werden. Dabei können beliebige Funktionsnamen, sie dürfen nur keine BASIC Schlüsselwörter sein, verwendet werden.

```

10 PRINT "GIB MIR DREI ZAHLEN";
20 INPUT A, B, C
30 PRINT "DIE SUMME DER DREI ZAHLEN IST";
40 PRINT FNSUM (A, B, C)
50 END
100 DEF FNSUM (X, Y, Z) = X + Y + Z

```

Unser Beispiel zeigt die Verwendung der Anweisung DEF in der Programmzeile 100 wird die Funktion FNSUM definiert. Sie hat drei Parameter, die in der Definition als X, Y und Z bezeichnet werden. Der Funktionswert, den diese selbst definierte Funktion, wie auch jede andere Funktionsform beim Aufruf bereitstellen muß, errechnet sich aus der Summe der drei Parameter.

Beim Aufruf der Funktion in der Zeile 40 unseres Beispiels können die Parameter beliebig bezeichnet werden, sie müssen nur, sollte dies von Bedeutung sein, in der richtigen Reihenfolge angeordnet werden.

### „JEDEM TIERCHEN SEIN PLÄSIERCHEN“

So, jetzt hätten wir eigentlich genug geschuftet. Wie wär's mit etwas Muse? Vielleicht etwas Duftendes zum Knabbern? Nun, wenn's magentfreundlich sein soll, dann muß es aber liebevoll gekocht sein. Okay, her mit dem Kochbuch. Natürlich dem „Elektronischen“. Denn wenn schon, dann setzen wir doch unseren COMPUTER ein. Na ja, wenn's einfach geht und praktisch ist, dann haben wir doch nichts dagegen, stimmt's? Es geht doch wirklich spielend, wenn wir das Kartesystem, oder besser DATEI-System „DATABASE“ verwenden. Mit Start des Programmwerkzeuges erscheint schon der erste Speisezettel – auch MENÜ genannt, der Sie geduldig durch das Angebot unseres COMPUTERS führt.

#### DATABASE MAIN MENU

- 1 Create a new file
  - 2 Examine/Update an existing file
  - 3 Add a field to a file
  - 4 Change field sizes/names
  - 5 Convert a file to text
  - 6 Make a View macro file
- Your selection –

Was wird denn da alles angeboten? Als Vorspeise vielleicht eine kleine Karteistruktur gefällig? Oder als Nachtisch zum Schluß eine Etikettenerstellung? Nein, wir greifen gleich einen der Hauptgänge heraus und tippen die 2. Der Schirm löscht sich, es erscheint die Meldung



## LOAD DATAFILE

What is the name of the

datafile you wish to load?

Wie hieß die Rezepte-Datei denn? Ich hab's – REZEPTE natürlich, ist doch sonnenklar. Also rasch eingetippt und der Schirm löscht sich schon wieder – es erscheint die Meldung „Bitte warten“. Irgendwoher kenn ich das. Auch egal, da füllt sich schon der Schirm mit dem ersten Rezept

## REZEPTE RECORD 1 PAGE 1/2

REZEPT Gefüllte Eier mit Schinken

RUBRIK Vorspeise

ZUTATEN 200 g Schinken, 120 g Butter  
6 Eier, Salz  
Spargel

ZUBERTG Schinken zweimal durch die  
Fleischmaschine, mit Butter  
und Dottern von 6 hartgek.  
Eiern vermischen, salzen,  
Eihäufchen füllen und mit  
Schinkenröllchen und  
Spargel garnieren

DATUM 84-03-26

LIEB.GR. Werthner, Vater, Robert

Nun ja, das sieht doch ganz köstlich aus, finden Sie nicht? Ob Sie's glauben oder nicht, es läßt sich doch ein jeder „einkochen“. Ich sag Ihnen, also bei meiner letzten Einladung war da eine nette junge Dame dabei, die war sichtlich von meinen Kochkünsten angetan. Ob's nur meine Kochkünste waren? Nun ja, das ist ein anderes Kapitel. Was kochte ich denn damals – jetzt fällt's mir nicht ein. Na, macht nichts, ich werde meinen treuen Partner befragen, meinen COMPUTER. Ein Blick auf die „FUNKTIONSTASTEN“, das sind die roten Tasten am „Keyboard“, zeigt mir, daß die Funktionstaste F2 mit SEARCH SUBSET gekennzeichnet ist. SEARCH heißt suchen und das wollen wir doch. Also flugs getippt und der Schirm meldet

## SEARCH REZEPTE

REZEPT	<input type="text"/>
RUBRIK	<input type="text"/>
ZUTATEN	<input type="text"/>
ZUBERTG	<input type="text"/>
DATUM	<input type="text"/>
LIEB.GR.	<input type="text"/>

Okay, das ist die leere Maske der Rezepte-Datei. Wieso Maske, meinen Sie? Nun ja, für so ein „Formular“ hat sich unter COMPUTER-Freaks eben der Begriff „Maske“ eingebürgert. Mit der Taste F2 fahren wir ins Feld „LIEB.GR.“ – da sind alle jene vermerkt, die auf das angegebene Gericht scharf sind und tippen \*Susi\* ein. Warum, fragen Sie? Ganz einfach, da ich nicht genau weiß, ob ich auch den Vornamen des Fräulein Susi in der Datei sehen habe. Und da mein COMPUTER so furchtbar pedant ist – er sucht nur nach Angaben, die exakt mit der Vorgabe übereinstimmen – signalisiere ich IHM mit den Sternchen bei \*Susi\*, das bei dem Wort Susi vielleicht etwas davor oder dahinterstehen könnte. Wichtig ist nur, ER sucht alle Rezepte heraus, die die Lieblingsgerichte von Susi sind. Gesagt, getan. Der Schirm löscht sich und meldet

## SEARCH COMPLETE

Records found =1

Press the SPACE BAR to continue

Nichts leichter als das! Mit leichtem Antippen an der Leertaste erscheint das erste Rezept

## REZEPTE Record 144 Page 1/2

REZEPT	Tirolerknödel
RUBRIK	Fleisch, Wurst
ZUTATEN	6 Semmeln, 4 Esslöffeln Mehl 1/4l Milch, 40 g Fett 180 g Wurst od. Schinken 2 Eier, Petersilie, Salz Zwiebel
ZUBERTG	Brot und Wurst würfelig schneiden, angeröstete Zwiebel dazugeben, mit Mehl und den anderen Zutaten vermischen und anziehen lassen Knödel formen, 10-15 Minuten ins kochende Wasser (+Salz) oder in Suppe ziehen lassen
DATUM	84-03-26
LIEB.GR.	Gertrud, Susi, Oma

Also, wenn einem da das Wasser nicht im Munde zusammenläuft? Und das nicht nur wegen dem Fräulein Susi!

## Dateneingabe

Aus dem Hauptmenü muß die Funktion

2 : EXAMINE/UPDATE AN EXISTING FILE  
aufgerufen werden.

Diese Funktion wird beim Arbeiten mit bestehenden Dateien am häufigsten verwendet. Sie ermöglicht es, neben der Dateneingabe auch noch Daten zu suchen sowie zu verändern.

Die Funktionen des Datenbanksystems werden über die Funktionstasten angewählt. Neben der DATABASE-Diskette enthält DATABASE auch noch eine Schablone, die die Belegung der Funktionstasten zeigt.

CTRL ▶											
SHIFT ▶	NEW DATA	SAVE SUBSET	GO TO RECORD				BACK FILE				
Database	INPUT DATA	LOAD FILE	SEARCH SUBSET	EDIT RECORD	START SUBSET	SORT FILE	PRINT SUBSET	PRINT SUBSET	OS 1 COMMAND	EXIT	

Die Funktion INPUTDATA, zur Eingabe von Daten in eine bereits definierte Datei, wird durch Drücken der Funktionstaste „FO“ aufgerufen. Mit Hilfe dieser Funktion werden freie Records beschrieben und an das Ende einer bereits bestehenden Datei angefügt. Nach dem Funktionsaufruf zeigt der Bildschirm ein leeres Record, das die Struktur entsprechend der Dateidefinition zeigt. In der ersten Zeile des Bildschirms wird die aktuelle Recordnummer angezeigt. Wenn NEW DATA für ein noch unbeschriebenes File aufgerufen wird, zeigt diese Zeile Record 1. Am Beginn des ersten Informationsfeldes wird ein weißer Block dargestellt. Dieser Cursor bewegt sich nach dem Drücken der Return-Taste, nachdem ein Feld beschrieben wurde, zum nächsten Feld. Die Cursortasten können zur Anwahl eines Feldes, das man in einem Record beschreiben möchte, verwendet werden.

DATABASE übernimmt jede Information die über Tastatur in ein angewähltes Feld eines Records eingegeben wird. Wenn mehr Zeichen eingegeben werden als durch die Recorddefinition vorgesehen, ertönt ein Warnsignal.

Während der Eingabe von Daten in ein Record, können Irrtümer, die bei der Eingabe passiert sind, entweder durch Drücken der „DELETE-Taste“ ausgebessert werden, oder der Cursor mit Hilfe der Cursor-Steuertasten auf die entsprechende Stelle gestellt und die Korrektur durch Übertippen durchgeführt werden.

Die Funktion „NEW DATA“ arbeitet ähnlich der Funktion „INPUT DATA“. NEW DATA bewirkt jedoch zusätzlich, daß DATABASE einen eventuell bereits existierenden Datei-Inhalt löscht und ihn durch die neu eingegebenen Informationen ersetzt.

Wird eine Funktionstaste irrtümlich betätigt, kann aus jeder Funktion durch Drücken der Taste „ESCAPE“ ein Abbruch durchgeführt werden.

Nach erfolgter Eingabe eines Records erscheint das leere Recordformular neuerlich, um die Eingabe von weiteren Records zu ermöglichen. Der Anwender braucht DATABASE nicht speziell anzuweisen die eingegebenen Informationen abzulegen, sie werden Record für Record abgespeichert.

### Informationsausgabe

Um den Inhalt der Records, die eine Datenbank bereits enthält, am Bildschirm anzusehen, kann zwischen den Records „hin- und hergeblättert“ werden. Dazu werden spezielle Tasten verwendet. Um zum ersten oder zum letzten Record eines Files zu gelangen, ist ebenfalls nur ein einfacher Tastendruck erforderlich. Ein bestimmtes Record kann durch die Funktion „GO TO RECORD“ unter Angabe der entsprechenden Recordnummer aufgesucht werden.

### Suchen in Records

Durch die Funktion „SEARCH SUBSET“ können Records mit bestimmten Eigenschaften aus der Datei ausgewählt werden. Nach Funktionsaufruf erscheint ein leeres Record am Bildschirm, wie es auch bei der Informationseingabe gezeigt wird. Der Anwender kann in beliebig viele Felder Eintragungen vornehmen, die bei Ausführung der Funktion als Suchkriterium verwendet werden. Wird in ein Feld eine bestimmte Zeichenkombination eingeschrieben und der Suchvorgang ausgelöst, sucht DATABASE alle Records der Datei nach Übereinstimmung mit den eingegebenen Zeichen ab. Es wird nach Durchführung der Suche ein Teil des Datenbestandes ausgewählt, der dem entsprechenden Suchkriterium entspricht. Wird beispielsweise in einer Namensdatei der Name Meier als Suchkriterium eingegeben und der Suchvorgang ausgelöst, enthält die Datei nach diesem Suchvorgang nur mehr alle jene Records, deren Namensfeld das Wort „Meier“ enthält.

Diese Auswahl ist temporär. Nach dem Ausschalten des Gerätes und neuerlichem Aufruf der Datenbank, steht wieder die gesamte Datenbank zur weiteren Bearbeitung zur Verfügung. Die gesamte Datei kann auch durch Aufruf der Funktion „WHOLE FILE“ neuerlich zur Bearbeitung in Zugriff gebracht werden.

Die Funktion „INVERT SUBSET“ bringt nach Durchführung eines Suchvorganges den gerade nicht ausgewählten also den inversen Teil der Datei zur Bearbeitung.

Zur Durchführung einer effektiven Auswahl sind spezielle Sonderzeichen vorgesehen. Der wichtigste davon ist der Stern (\*). Wird als Vergleichszeichen ein Stern angegeben, wird der Vergleich dieses Zeichens mit jeder Anzahl beliebiger Zeichen als positiv gewertet. Daraus ergeben sich beispielsweise nachstehende Übereinstimmungen.

BROWN	findet	BROWN
	findet nicht	BROWNING
	oder	BROW
	oder	ROWN
	oder	NUTBROWN
BROWN	findet nicht	BR OWN
	oder	BROWN
BROWN*	findet	BROWN
	oder	BROWNS
	oder	BROWNING
	oder	BROWN AND POULSON
*BROWN	findet	BROWN
	oder	H. A. BROWN
	oder	GEORGE BROWN
*BROWN*	findet	BROWN
	oder	BROWNING
	oder	H. A. BROWN
	oder	SMITH, BROWN AND JONES

### Sortieren

Die Funktion „SORT FILE“ dient dazu, eine sortierte Reihenfolge der Records herzustellen. Nach Aufruf der Funktion wird die leere Recordstruktur am Bildschirm angezeigt. Der Anwender kann dann durch Eingabe von Ziffern die Priorität der einzelnen Felder beim Sortiervorgang festlegen. Das Feld, das mit der Priorität 1 bezeichnet ist, wird für die primäre Sortierreihenfolge herangezogen. Werden Felder gleichen Inhalts angetroffen, wird für die Reihenfolge innerhalb dieser Records das Feld mit der nächsten Priorität herangezogen.

Es muß nicht jedes Feld mit einer Sortier-Priorität versehen sein. Der Sortiervorgang kann entweder entsprechend einer numerischen Reihenfolge oder dem alphabetischen Inhalt eines Feldes ausgeführt werden.

### Abspeichern von Teilinformationen

Nach Durchführen von Sortier- oder Auswahlvorgängen steht die entsprechende Auswahl der Datei nur temporär zur Verfügung. Soll diese Auswahl abgespeichert werden, muß die Funktion „SAVE SUBSET“ aufgerufen werden. Mit dieser Funktion kann der ausgewählte oder sortierte Teil der Datei unter einem wählbaren Dateinamen am Externspeichermedium abgelegt werden.

Spezielle Funktionen stehen zum Ausdruck von Records oder einzelner Felder der Records zur Verfügung. Der einfachste Weg, einen ausgewählten Teil der Datei auszu-drucken besteht darin, durch Ausführung der Funktion „SEARCH SUBSET“ eine Selektion vorzunehmen und diesen ausgewählten Teil der Datei durch die Funktion „PRINT SUBSET“ am Drucker auszugeben. Nach Aufruf der Funktion „PRINT SUBSET“ kann in ein leeres Recordformular eingetragen werden, welche Felder in welcher Reihenfolge am Drucker ausgegeben werden. Damit ist eine einfache Druckformatierung möglich.

DATABASE erlaubt dem Anwender auch die Änderung einer Dateistruktur bei bereits eingegebenen Record-Inhalten. Das bedeutet, daß auch die Struktur einer bereits bestehenden Datei, beispielsweise durch Hinzufügen oder Verändern eines weiteren Feldes im Recordaufbau geändert werden kann.



**FOLGE 11 „Ein COMPUTERSPIEL“**

Heute werden Sie eine Überraschung erleben. Wir haben mit unserem COMPUTER ein Spiel vorbereitet. Nichts außergewöhnliches, sagen Sie? Prinzipiell haben Sie recht, aber dieses Spielprogramm haben wir selber programmiert. Im Schweiße unseres Angesichts! Wir haben's „HANGMAN GAME“ genannt. Ich werde es Ihnen kurz erklären. Zuerst fragt Sie der COMPUTER via Bildschirm nach einem zu erratenden Wort. Natürlich muß das jemand anderer eintippen, denn Sie können doch Ihr eigenes Wort nicht erraten – nein, das wäre ein Schwindel. Also, nach der Eingabe des Wortes zeichnet der COMPUTER den Galgen am Bildschirm und gibt darunter so viele Bindestriche aus, als Buchstaben im zu erratenden Wort enthalten sind.



```

*****
      <>
      ##
     ####
     ####
     ##
*****
# #####

```

```

*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****

```

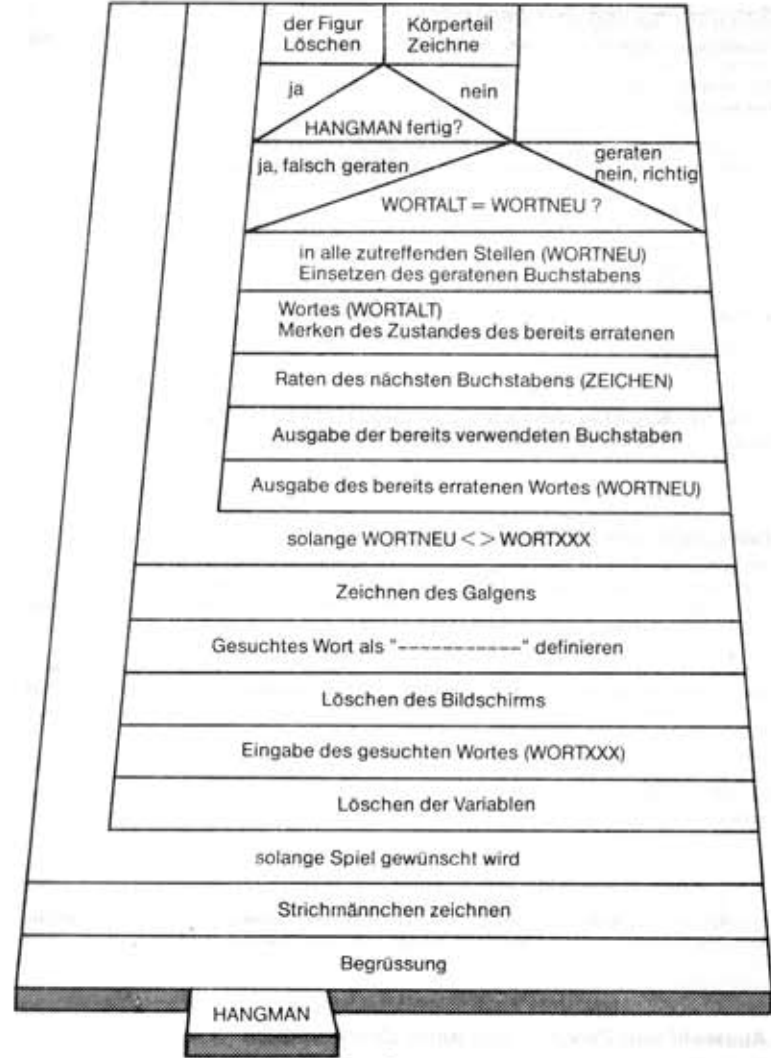
```
*****
```

```

B -- N -- F      QXWR
RATE EINEN BUCHSTABEN:

```

So, und jetzt geht's erst richtig los für Sie. Sie müssen die einzelnen Buchstaben erraten, die wenn sie im Wort enthalten sind, an der entsprechenden Stelle in der Zeichenkette angezeigt werden. Sie haben allerdings nur eine beschränkte Anzahl von Rateversuchen zur Verfügung. Bei jedem Fehlversuch wird ein Teil unseres armen Delinquenten am Galgen gezeichnet – Kopf, Arme, Hände, usw. Wenn der Galgen fix und fertig ist bevor Sie das gesamte Wort erraten haben, dann haben Sie das Spiel verloren. Aber unser COMPUTER ist fair. Denn am Schluß werden Sie zu einem neuerlichen Versuch eingeladen. Nun, haben Sie's erraten? Ich verrate es Ihnen, das gesuchte Wort heißt Bahnhof. Wie ich das gemacht habe? Nun, das war ganz einfach – ein bißchen STRUKTOGRAMM und ein paar BASIC-Zeilen – und schon wars fertig. Glauben Sie nicht? Warten Sie, ich zeige es Ihnen. Wo habe ich denn die Skizze nur? Sie müssen verzeihen, ein COMPUTER-Freak – wenn der so mitten in Aktion ist, der schaut nicht links und rechts. Da hab' ichs schon, sehen Sie



Das ist doch völlig durchsichtig, stimmt's? Haben Sie die Programmschleifen erkannt? Das sind die zurückführenden Bahnen im Struktogramm. Beispielsweise wird der rechts unten stehende Gesamtblock „solange“ wiederholt, bis das erratene Wort ident mit dem vorgegebenen ist.

## Behandlung von Zeichenketten

Zahlen und numerische Variable können mit Hilfe der Rechenoperationen und arithmetischen Funktionen verknüpft werden. Wörter oder andere Zeichengruppen, die in Zeichenketten Variablen gespeichert werden, können mit speziellen Funktionen manipuliert werden.

Die einfachste Manipulation ist das Zusammenfügen von Zeichenketten. Dabei wird das Pluszeichen verwendet, das jedoch eine andere Bedeutung als bei arithmetischen Operationen hat. Subtraktions- oder Multiplikationszeichen können daher nicht analog verwendet werden.

```
10 A$="HAUPT"
20 B$="BAHNHOF"
30 C$=A$+B$
40 PRINT C$
> RUN
HAUPTBAHNHOF
>
```

In unserem Beispiel werden die String Variablen A\$ und B\$ in der Zeile 30 verknüpft und der Variablen C\$ zugeordnet. Sie enthält die Aneinanderfügung der beiden Inhalte von A\$ und B\$.

## Die Länge von Zeichenketten

Wenn Zeichenketten in Zeichenkettenvariablen gespeichert werden, ist häufig die Anzahl der dann in der Variablen enthaltenen Zeichen unbekannt. In vielen Programmen ist es aber notwendig zu wissen, wieviele Zeichen eine Zeichenkettenvariable in einem bestimmten Betriebszustand beinhaltet.

Die numerische Funktion LEN kann dazu benützt werden, um die Anzahl der Zeichen in einer Folge zu zählen. Sie gibt die Länge der Zeichenketten in einer Zeichenkettenvariablen wieder.

```
10 INPUT A$
20 X=LEN(A$)
30 PRINT A$, "HAT", X, "ZEICHEN"
> RUN
? AUTO
AUTO HAT 4 ZEICHEN
>
```

In unserem Beispiel wird der String Variablen A\$ in der Programmzeile 10 über die Tastatur eine Zeichenkette zugeordnet. Wir geben beim Programmablauf das Wort AUTO ein. In der Zeile 20 wird die Länge der Zeichenkette, in unserem Beispiel 4, der Variablen X zugeordnet.

## Auswahl von Zeichen aus einer Zeichenkette

Bei der Manipulation von Zeichenketten taucht häufig das Problem auf, aus einer Zeichenkette einzelne Zeichen auszuwählen. Dazu stehen die Funktionen LEFT\$, RIGHT\$, MID\$ zur Verfügung.

Die Funktion LEFT\$ erlaubt es, eine wählbare Anzahl von Zeichen aus einer angegebenen Zeichenkette auszulesen. Dabei wird am Anfang, also am „linken Ende“ der Zeichenkette begonnen.

```
10 A$="HAUPTBAHNHOF"
20 B$=LEFT$(A$, 5)
30 PRINT B$
> PRINT
HAUPT
```

In unserem Beispiel haben wir der String Variablen A\$ das Wort HAUPTBAHNHOF zugeordnet. Durch die Funktion LEFT\$ in der Zeile 20 ordnen wir der Variablen B\$, fünf Zeichen beginnend vom linken Ende der String Variablen A\$ zu. B\$ wird daher die Zeichenkette HAUPT zugeordnet.

Ähnlich wie LEFT\$ arbeitet die Funktion RIGHT\$. Bei Verwendung der Funktion RIGHT\$ allerdings wird vom Ende der Zeichenkette ausgegangen.

```
10 A$="HAUPTBAHNHOF"
20 B$=RIGHT$(A$, 3)
30 PRINT B$
```

```
> RUN
HOF
```

In unserem Beispiel werden der Variablen B\$ in der Zeile 20, aus der Zeichenkettenvariablen A\$, drei Zeichen beginnend vom rechten Ende der Zeichenkette zugeordnet. B\$ erhält daher die Zeichenkette HOF.

Die Funktion MID\$ erlaubt es darüber hinaus, eine beliebige Stelle in der Zeichenkette zu bestimmen, bei der mit der Übertragung der Einzelzeichen begonnen wird. Mit dieser Funktion können daher beliebige Zeichen, auch aus der Mitte der Zeichenkette ausgelesen werden.

```
10 A$="HAUPTBAHNHOF"
20 B$=MID$(A$, 6, 4)
30 PRINT B$
```

```
> RUN
BAHN
```

In unserem Beispiel wird in der Zeile 20 der Zeichenkettenvariablen B\$ ein Teil aus der Zeichenkettenvariablen A\$ zugeordnet. Dabei wird angegeben, daß ab dem sechsten Zeichen vier Zeichen zugeordnet werden sollen. Daher wird der Variablen B\$ die Zeichenkette BAHN zugeordnet.

## Graphische Darstellungen

Mit dem Acorn B Mikrocomputer und dem Electron kann zur Darstellung von Text und Graphiken auf dem Bildschirm aus acht, mit dem Electron aus sieben unterschiedlichen Betriebsarten gewählt werden.

Diese Betriebsarten dienen einerseits zur Textdarstellung mit unterschiedlicher Anzahl von Zeichen am Bildschirm, sowie zur Graphikdarstellung mit unterschiedlicher Auflösung und Anzahl der verfügbaren Farben. Der Acorn B Mikrocomputer hat zusätzlich zum Electron eine Betriebsart die speziell auf TELETEXT ausgerichtet ist, sie soll hier nicht beschrieben werden.

Der Bildschirmmodus kann jederzeit durch Eingabe des Kommandos

MODE X

gewechselt werden, wobei X einen Wert zwischen 0 und 7 annehmen kann. Die möglichen 7 Betriebsarten ergeben sich aus der nachstehenden Aufstellung.

Modus	Graphik	Farben	Text
0	640x256	2	80x32
1	320x256	4	40x32
2	160x256	16	20x32
3	nur Text	2	80x25
4	320x256	2	40x32
5	160x256	4	20x32
6	nur Text	2	40x25

Der Modus 5 bietet beispielsweise eine mittlere graphische Auflösung und die Verwendung von 4 Farben. Wenn man diese Betriebsart durch Eingabe von Modus 5 auswählt, sind am Bildschirm nur 2 Farben sichtbar:

weiße Zeichen auf schwarzem Grund.

Durch die Anweisung COLOR kann die Textfarbe geändert werden. Dazu muß nach der Anweisung COLOR eine logische Farbnummer angegeben werden. Durch die Farbnummer selbst kann auch zusätzlich die Hintergrundfarbe angesprochen werden.

Vordergrundfarbe		Hintergrundfarbe	
Log. Nr.	Effektive Farbe	Log. Nr.	Effektive Farbe
<b>Modi 0, 3, 4, 6</b>			
0	schwarz (0)	128	schwarz (0)
1	weiß (7)	129	weiß (7)
<b>Modi 1, 5</b>			
0	schwarz (0)	128	schwarz (0)
1	rot (1)	129	rot (1)
2	gelb (3)	130	gelb (3)
3	weiß (7)	131	weiß (7)
<b>Modus 2</b>			
0	schwarz (0)	128	schwarz (0)
1	rot (1)	129	rot (1)
2	grün (2)	130	grün (2)
3	gelb (3)	131	gelb (3)
4	blau (4)	132	blau (4)
5	violett (5)	133	violett (5)
6	türkis (6)	134	türkis (6)
7	weiß (7)	135	weiß (7)
8	blink. schwarz/weiß (8)	146	blink. schwarz/weiß (8)
9	blink. rot/türkis (9)	137	blink. rot/türkis (9)
10	blink. grün/violett (10)	138	blink. grün/violett (10)
11	blink. gelb/blau (11)	139	blink. gelb/blau (11)
12	blink. blau/gelb (12)	140	blink. blau/gelb (12)
13	blink. violett/grün (13)	141	blink. violett/grün (13)
14	blink. türkis/rot (14)	142	blink. türkis/rot (14)
15	blink. weiß/schwarz (15)	143	blink. weiß/schwarz (15)

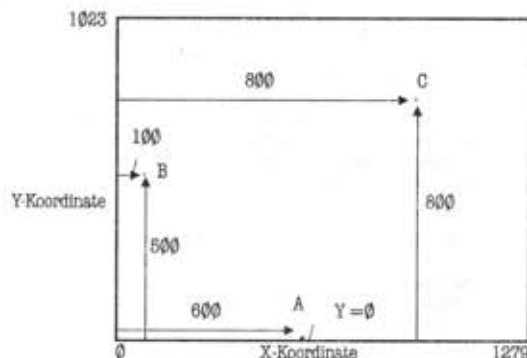
## Graphiken

Beim Erstellen von Graphiken in den graphischen Betriebsarten können spezielle Anweisungen des Acorn B Mikrocomputers verwendet werden. Durch die Anweisung DRAW können Linien am Bildschirm gezogen werden, wobei die Linien jeweils vom zuletzt angesprochenen Koordinatenpunkt zu dem mit der Anweisung DRAW angegebenen Koordinatenpunkt gezogen wird. Durch die Anweisung MOVE kann ein Koordinatenpunkt aufgesucht werden ohne daß dabei ein Linienzug entsteht.

Durch die Anweisungen

```
MOVE 100, 500
DRAW 800, 800
```

wird eine Linie zwischen den Punkten B und C am Bildschirm gezogen.



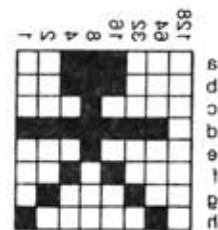
## Bewegte Trickdarstellung

Figuren können auf zwei Arten definiert und am Bildschirm bewegt werden.

Die naheliegende Möglichkeit besteht darin, eine Prozedur zu erstellen, in der das darzustellende Objekt, z. B. ein Ball, auf dem Bildschirm gezeichnet wird. Diese Prozedur kann aus einem Hauptprogramm aufgerufen werden. Die Absolutposition der Figur am Bildschirm wird bei dieser Prozedur als Parameter übergeben. Das aufrufende Hauptprogramm kann die Prozedur beispielsweise in einer Schleife mit laufendem Parameter aufrufen und so den Ball von einem Ende des Bildschirms zum anderen bewegen.

Bewegte Graphiken können aber auch mit Hilfe benutzerprogrammierbarer Graphikzeichen erstellt werden. Jedem Zeichen, das über die Tastatur eingegeben werden kann und das am Bildschirm dargestellt wird, ist ein ASCII Code zugeordnet. Bei der Aufgabe, ein derartiges Zeichen am Drucker oder am Bildschirm darzustellen, sucht der Computer den jeweiligen Code in einer Tabelle und schreibt dann das entsprechende Zeichen in Form einer 8x8 Punktmatrix. Ein Buchstabe A hat z. B. den Code 65. Gewisse ASCII Werte kann der Benutzer jedoch noch selbst definieren, sie sind durch Standardsymbole nicht belegt.

Dies sind die unbesetzten Werte von 224 bis 255. Sie können mit Hilfe der VDU 23 Anweisung definiert werden. Wenn ein derartiges Symbol in einer 8x8 Matrix festgelegt ist, kann es wie ein ASCII Zeichen, also wie ein Buchstabe mit ihrem Code aufgerufen und an jede beliebige Stelle des Bildschirms gestellt werden.



In unserem Beispiel haben wir eine kleine Figur in einer 8x8 Matrix dargestellt. Diese Figur kann wie ein ASCII Zeichen angesprochen und z. B. mit Hilfe der PRINT Anweisung dargestellt werden. Durch zusammenlegen mehrerer 8x8 Blöcke können auch größere Figuren definiert werden.

Ich sollte ein paar Zeilen schreiben. Nun, nichts leichter als das! Hingesetzt und ein Blatt vollgekrizelt. Können vor lauter Lachen. Erstens ist just grad jetzt kein Blatt zu greifen und der Kugelschreiber wollte schon beim letzten Mal nicht so recht. Halt, ich hab doch noch ein Programm, das ich noch nie ausprobiert habe. Es heißt „VIEW“ oder so und soll irgend so eine Textverarbeitung sein. Was das mit meinen Zeilen zu tun haben soll? Sie meinen, ich soll nicht so viel fragen und lieber das Programm ausprobieren? Gut, werd' ich tun. Sie meinen gleich? Also, wenn Sie mich dabei begleiten, dann tu ich's. Zuerst mache ich's mir bequem vor dem COMPUTER, dann IHN eingeschaltet und das Programm gestartet. Der Schirm wechselt die Farbe und meldet sich mit

VIEW A2.1

No text  
Editing No File  
Screen Mode ?  
Printer default



So, das ist alles. Ein paar englische Zeilen. Was sagen Sie, ich soll die ESCAPE-Taste rühren! Auf Ihre Verantwortung. Und siehe da, der Schirm wechselt sein Aussehen zu

```
FJ .....*.....*.....*.....*.....*.....*.....*.....*.....*.....<
```

Bis auf den blinkenden weißen Strich, den Cursor, reißt mich das Bild nicht unbedingt vom Hocker. Sie meinen, ich soll nun etwas tippen? Ja, was soll ich da in der Schnelle schreiben? Da fallen mir ein paar Worte von Eugen Roth ein. Wie war das noch, ich glaube so

„Ein Mensch schreibt mitternächtigt tief an die Geliebte einen Brief, der schwuel und voller Nachtgefühl. Sie aber kriegt ihn morgenkuehl, liest gaehndend ihn und wirft ihn weg. Man sieht, der Brief verfehlt den Zweck. Der Mensch, der nichts mehr von ihr hoert, ist seinerseits mit Recht empoert und schreibt am hellen Tag gekraenkt und saugrob, was er von ihr denkt. Die Liebste kriegt den Brief am Abend, soeben sich entschlossen habend, den Menschen dennoch zu erhoeren – der Brief muss diesen Vorsatz stoeren. Nun schreibt, die Grobheit abzubitten, der Mensch noch einen zarten dritten und vierten, fuerften, sechsten, siebten der herzlos schweigenden Geliebten. Doch bleibt vergeblich alle Schrift, wenn man zuerst daneben trifft.“

Kannten Sie die Zeilen? Elefant im Porzellanladen, stimmt's? Jetzt werde ich Ihnen einmal zeigen, was so eine Textverarbeitung alles kann. Wir wollen nun den Block formatieren. Was das heißt? Ganz einfach, der rechte Rand des Gedichtes wird automatisch nach rechts gezogen. Wie das geht? Nun, mit einer der vielen Funktionstasten – Sie wissen schon, das sind die roten Tasten – können Sie das Kommando „FORMAT BLOCK“ geben und flugs sieht das Gedicht so aus

„Ein Mensch schreibt mitternächtigt tief an die Geliebte einen Brief, der schwuel und voller Nachtgefühl. Sie aber kriegt ihn morgenkuehl, liest gaehndend ihn und wirft ihn weg. Man sieht, der Brief verfehlt den Zweck. Der Mensch, der nichts mehr von ihr hoert, ist seinerseits mit Recht empoert und schreibt am hellen Tag gekraenkt und saugrob, was er von ihr denkt. Die Liebste kriegt den Brief am Abend, soeben sich entschlossen habend, den Menschen dennoch zu erhoeren – der Brief muss diesen Vorsatz stoeren. Nun schreibt, die Grobheit abzubitten, der Mensch noch einen zarten dritten und vierten, fuerften, sechsten, siebten der herzlos schweigenden Geliebten. Doch bleibt vergeblich alle Schrift, wenn man zuerst daneben trifft.“

Sieht doch aus wie aus der Druckerei, stimmt's? So, jetzt noch schnell auf der Diskette abgelegt. Das geschieht mit dem Kommando „SAVE“, dabei muß ich noch eine Namen für den Text finden – ich will ihn ROTHnennen und schon schnurrt das Laufwerk. Wenn ich jetzt den COMPUTER ausschalte, dann ist mein schönes Gedicht nicht verloren. Ich kann's jederzeit mit dem Kommando „LOAD“ und der Angabe des Namens ROTH wieder in den Speicher des COMPUTERS holen. Wie, Sie hätten sich gerne das Gedicht mitgenommen? Auch möglich. Ich werde es Ihnen auf dem Drucker ausdrucken. Okay? Zum Schluß darf ich Ihnen noch verraten, daß dieses Büchlein zur Gänze auf diesem COMPUTER mit dem Programmwerkzeug „VIEW“ (Textverarbeitung) geschrieben wurde. Hätten Sie nicht geglaubt?

### Textverarbeitung

Textverarbeitungsprogramme dienen zur Unterstützung der Eingabe und Verwaltung von Zeichen in einem Computersystem. Mit Hilfe eines Textverarbeitungsprogrammes, das als Softwarewerkzeug entweder auf Diskette oder als Steckmodul geliefert wird, können am Bildschirm eines Computers Schriftstücke erstellt werden.

Der wesentliche Unterschied der Erstellung von Schriftstücken mit Hilfe eines Textverarbeitungs-Systems im Gegensatz zur Verwendung einer modernen Schreibmaschine besteht darin, daß der eingegebene Text nicht sofort auf Papier gedruckt wird sondern vorerst elektronisch gespeichert bleibt.

Der elektronisch gespeicherte Text kann am Bildschirm angesehen, korrigiert und ergänzt werden. Dies ist deshalb so einfach möglich, weil der Computer den elektronisch gespeicherten Text sehr einfach überschreiben, verschieben und verändern kann. Dies ist bei gedrucktem Text nicht mehr möglich.

Der Anwender kann Schriftstücke, die im Arbeitsspeicher des Computers zur Bearbeitung vorliegen auch auf das Externspeichermittel ablegen. Dort speichert die Dateiverwaltung den Text als File, er kann für spätere Bearbeitungsvorgänge wiederum einfach in den Arbeitsspeicher zurücktransferiert werden.

Textverarbeitungsprogramme bieten neben den einfachen Eingabe-, Korrektur- und Druckmöglichkeiten zusätzliche Funktionen, die das Schreiben von Texten erleichtern.

Vor allem die Möglichkeit, bereits abgespeicherte Texte und Textteile zu einem gerade eingegebenen Text hinzuzufügen, ermöglicht die Verwendung von Standardtextblöcken. Ein gutes Beispiel dafür sind die Standard-Verabschiedungssätze in einem Brief. Sie können in einer Datei gespeichert werden und brauchen anstelle jedesmal neu geschrieben, nur mehr von dort an den gerade in Erstellung befindlichen Brief angefügt zu werden.

Sehr wichtige Funktionen der Textverarbeitung bestehen auch darin, ein bereits erstelltes Dokument zur Ausgabe vorzubereiten. Darunter versteht man unterschiedliche Formatie-

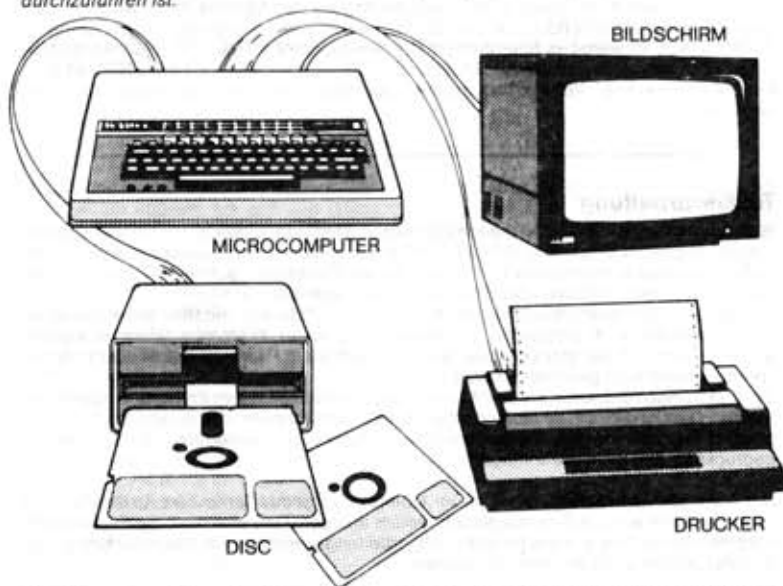
rungsvorgänge. Es können sowohl Zeilenbreite wie auch Absätze und Seitenwechsel auch im nachhinein vom Anwender definiert werden. Die Gestalt eines auch bereits geschriebenen Textes läßt sich damit im nachhinein an unterschiedliche Gegebenheiten beim Ausdruck einfach anpassen.

### Textverarbeitungsprogramm VIEW

Für den Acorn Mikrocomputer steht das Textverarbeitungsprogramm „VIEW“ zur Verfügung. Es handelt sich dabei um ein Programm-Modul, das in einem Read-Only-Memory enthalten ist. Um den Acorn B Mikrocomputer mit diesem Standardprogramm auszurüsten, muß das mit VIEW mitgelieferte ROM in einen freien Stecksockel der Leiterplatte gesteckt werden. Je nachdem, welche Position im Acorn B Mikrocomputer gewählt wird, wird entweder VIEW oder BASIC oder ein anderes eingestecktes Programmwerkzeug automatisch beim Einschalten des Computers geladen. Darüber hinaus ist aber der Übergang zu allen eventuell eingesteckten Standardprogrammen durch Eingabe einfacher Kommandos möglich.

VIEW kann mit beiden, für den Acorn B Mikrocomputer vorgesehenen Externspeichermedium arbeiten. Das bedeutet, daß sowohl mit einem angeschlossenen Kassettenrecorder als Speicher oder einem Floppy Disk System, Textverarbeitung durchgeführt werden kann. Wenn beide Möglichkeiten vorgesehen sind, kann zwischen den beiden Externspeichermedien durch entsprechende Betriebssystem-Befehle hin- und hergeschaltet werden.

Es ist jedoch wichtig zu erkennen, daß Textverarbeitung mit all den Vorteilen der automatischen Verwaltung von Texten auf einem Kassettengerät nur mit großen Schwierigkeiten durchzuführen ist.

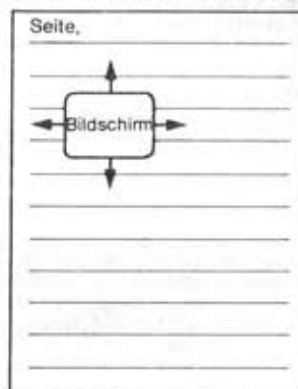


Nach dem Einschalten des Computers und Starten des Textverarbeitungsprogrammes befindet man sich im sogenannten Command-Mode. In dieser Betriebsart des Textverarbeitungs-Systems können Befehle an das Betriebssystem eingegeben werden. Wichtige Einstellungen, wie beispielsweise die des Bildschirm-Modus, können durchgeführt werden. Der Bildschirm zeigt gleichzeitig an, in welchem Bildschirm-Modus man sich befindet, ob ein Printer angeschlossen ist und welchen Text man gerade bearbeitet.

VIEW ist in allen Bildschirm-Betriebsarten des Acorn B Mikrocomputers funktionsfähig. Das bedeutet, daß man durch Auswahl des „Screen-Mode“ die Anzahl der Zeichen pro Zeile und damit auch die Zeichengröße bestimmen kann.

Für die Eingabe von Schriftstücken, die auf Seiten gedruckt werden sollen, ist der Screen-

Mode 3, der die Eingabe von 74 Zeichen in 24 Zeilen erlaubt, gut verwendbar.



Aus dem „Command-Mode“ kann durch Eingabe des Kommandos „NEW“ in den Textmode übergegangen werden. In dieser Betriebsart zeigt der Bildschirm einen Ausschnitt des bearbeiteten Textblattes. Der Bildschirm kann über dieses Textblatt bewegt werden und erlaubt die Eingabe von Zeichen an jeder beliebigen Stelle. Innerhalb des Bildschirms kann die Eingabestelle durch den Cursor ausgewählt werden. Der Cursor wird durch die Cursor-Steuertasten bewegt.

Die ersten Zeilen des Bildschirms zeigen die Tabulatorpositionen. Im Bildschirm-Mode 3 ist automatisch an jeder achten Position des Bildschirms ein, wie bei einer Schreibmaschine verwendbarer, Tabulatorpunkt gesetzt. Durch entsprechende Befehle von VIEW können diese Tabulator-Positionen auch individuell gesetzt werden.



Um die Auswahl aus den vielfältigen Funktionen von VIEW für den Anwender möglichst einfach zu gestalten, sind die Funktionstasten mit den Steuerfunktionen belegt. VIEW enthält neben der Bedienungsanleitung und dem Programm ROM auch eine Schablone, die die Belegung der Funktionstasten mit den einzelnen Funktionen zeigt.

CTRL ▶	DELETE BLOCK	NEXT MATCH	FORMAT MODE	JUSTIFY MODE	INSERT MODE	DEFAULT FILLER	SPLIT LINE	CONCATENATE LINES	MARK AS FILLER	
SHIFT ▶	MARK BLOCK	SWAP CASE	RELEASE MARGINS	DELETE UP TO CHARACTER	HIGHLIGHT 1	HIGHLIGHT 2	GO TO MARKER	SET MARKER	EDIT COMMAND	DELETE COMMAND
	FORMAT BLOCK	TOP OF TEXT	BOTTOM OF TEXT	DELETE END OF LINE	BEGINNING OF LINE	END OF LINE	INSERT LINE	DELETE LINE	INSERT CHARACTER	DELETE CHARACTER

### Befehle zur Texteingabe

Der Cursor kann durch Befehle, die mit Funktionstasten aufgerufen werden, an bestimmte Stellen des Textes gebracht werden. Die Funktion „TOP OF TEXT“ bewegt den Cursor an die Stelle des ersten Zeichens im Text.

Die Funktion „BOTTOM OF TEXT“ bewegt den Cursor an die Position hinter dem letzten eingegebenen Zeichen im Text. Der mit diesen Funktionen angesprochene Text kann natürlich größer als im Bildschirm angezeigt sein.

Die Funktion „BEGINNING OF LINE“ bewegt den Cursor zur ersten Position in der Eingabezeile. Die Funktion „END OF LINE“ bewegt den Cursor an die letzte Stelle der gegenwärtig bearbeiteten Zeile.

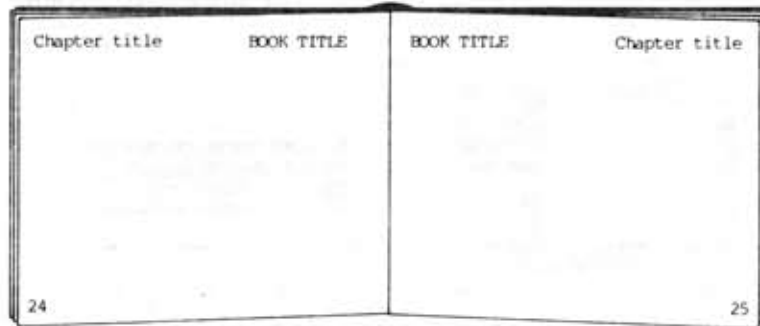
CTRL ▶	DELETE BLOCK	NEXT MATCH	FORMAT MODE	JUSTIFY MODE	INSERT MODE	DEFAULT FILLER	SPLIT LINE	CONCATENATE LINES	MARK AS FILLER	
SHIFT ▶	MARK BLOCK	SWAP CASE	RELEASE MARGINS	DELETE UP TO CHARACTER	HIGHLIGHT 1	HIGHLIGHT 2	GO TO MARKER	SET MARKER	EDIT COMMAND	DELETE COMMAND
	FORMAT BLOCK	TOP OF TEXT	BOTTOM OF TEXT	DELETE END OF LINE	BEGINNING OF LINE	END OF LINE	INSERT LINE	DELETE LINE	INSERT CHARACTER	DELETE CHARACTER

Weitere wichtige Befehle erlauben die Eingabe von zusätzlichen Zeilen im bereits geschriebenen Text sowie auch das Löschen von Zeilen. Gleichmaßen können auch einzelne Zeichen im bereits geschriebenen Text eingegeben oder ausgelöscht werden. Eine weitere Funktion löscht einen gesamten Block innerhalb eines Textes.

### Anweisungen im Text

Durch die Funktion „EDIT COMMAND“ können in den Text Anweisungen eingegeben werden. Diese Anweisungen bewirken beim Ausdruck des Dokuments keinen Zeichen- ausdruck sondern Steueranweisungen an den Drucker. Beispiele dafür sind Seiten- vorschub, das Setzen von Zeichensätzen oder Zeilen bzw. Zeichenbreiten für einen bestimmten Drucker.

Mit Hilfe von VIEW können beim Ausdruck auch Überschriften und eine automatische Seitennumerierung für jede ausgedruckte Seite generiert werden. Die Überschriftenzeilen bleiben für jede Seite gleich, die Seitennumerierung wird von VIEW automatisch weiterge- zählt. Durch eine spezielle Funktion kann die Ausgabe von Seiten für Bücher vorbereitet werden. Dabei tritt häufig der Wunsch auf, daß die Seitennumerierung jeweils einmal am linken äußeren und in der nächstfolgenden Seite am rechten äußeren Rand zu liegen kommt. Seitenüberschriften können für jede zweite Seite gespiegelt werden, um in einem aufgeblättern Buch einen symmetrischen Aufbau herzustellen.

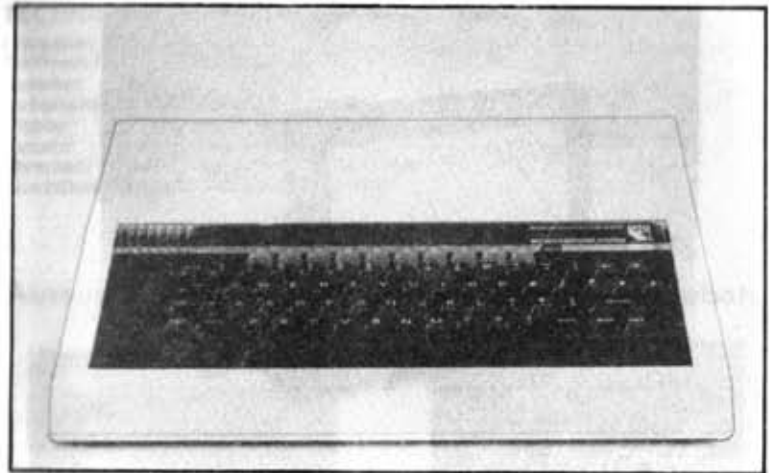


Die Anwendung von VIEW ist trotz der in diesem Programmwerkzeug gebotenen Funk- tionsvielfalt einfach. Die Zeit, die man benötigt, um mit VIEW effektiv arbeiten zu können, ist sicherlich nicht länger, als eine entsprechende Gewöhnungsphase für eine neue Schreibmaschine mit komplexen Funktionen. Besonders durch die Verwendung der Funk- tionstasten und der Funktionstasten-Schablone, können auch „Nicht-Computerprofis“ mit VIEW professionell Texte erstellen.

## FOLGE 13 AUCH DIE ZUKUNFT WIRD ES ZEIGEN!

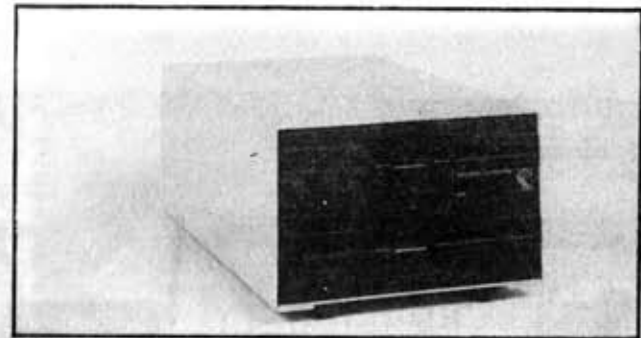
Acorn Computer mit der überlegenen Technik und der Intelligenz von morgen sind bereits schon heute so konzipiert, daß sie zukunftsorientiert sind.

Das heißt: der Acorn B Computer verfügt bereits in der Grundausstattung über eine Reihe von Schnittstellen, mittels derer die Technologie der kommenden Jahre angeschlossen werden kann, ohne einen Hardware-Wechsel vornehmen zu müssen.



### TECHNIK

64 K, erweiterbar, exzellente Farbgrafik (640x256=163.840 Punkte) – 16 Farben, 10 Funk- tionstasten für die freie Definition von Befehlen, Schreibmaschinentastatur mit 73 Tasten in der QWERTY-Anordnung.



### DOPPELLAUFWERK

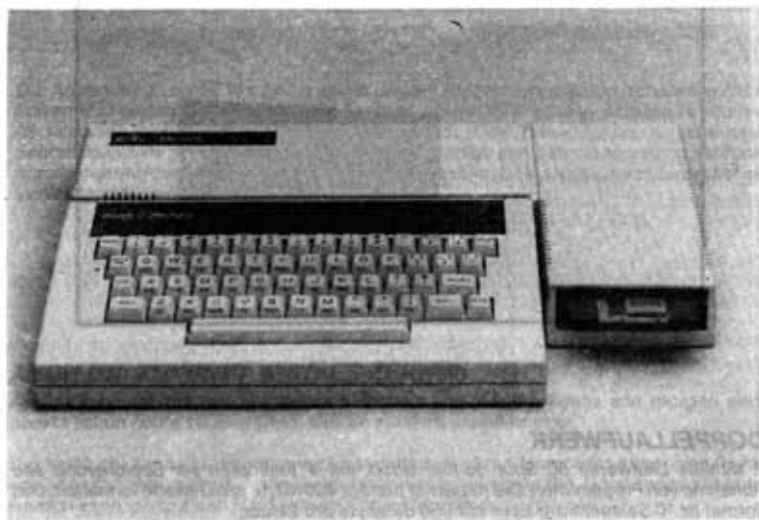
2 slimline Laufwerke 80 Spur double sided, mit 4 Tonköpfen zur Speicherung und Abnahme von Programmen. Die Kapazität beträgt 400 KByte pro Diskette formatiert. Das Format ist 10 Sektoren pro Spur mit 256 databyte pro Sektor.



**14" RGB-FARBMONITOR**

für höchste Ansprüche inkl. Verbindungskabel.

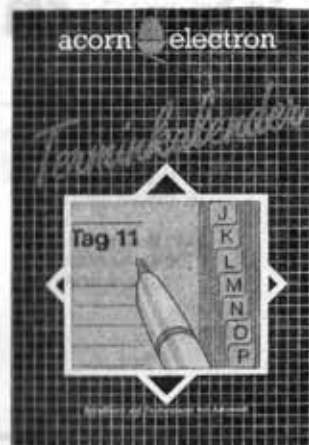
Der Acorn Electron wächst mit den Wünschen des Benützers. Der Anwender hat die Möglichkeit sich mit dem System zu entwickeln. So gibt es z. B. schon jetzt eine neue 3 1/2" Floppy-Disk, welche wegweisend für eine neue Speicher-Generation ist.

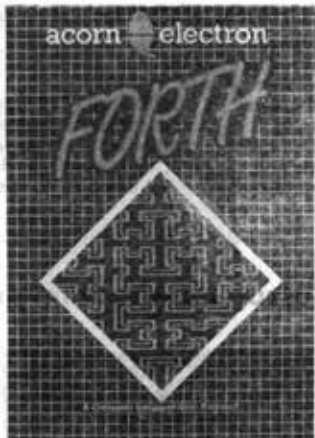


**TECHNISCHE DATEN für ACORN ELECTRON**

Prozessor: 6502-A  
 Taktfrequenz: 2 MHz  
 Speicher: 64 K  
 Farbgraphik: 640x256 = 163.840 Punkte  
 Display: 7 Modi mit verschiedener Auflösung  
 Tastatur: Schreibmaschinentastatur mit 56 Tasten davon 10 frei definierbar  
 Sprachen: BASIC- und Assembler-Interpreter  
 Anschlüsse: UHF/TV, Video, RGB, Cassette

**Auszug aus dem umfangreichen Software-Angebot:**





# Kaindl A. Berghofer 30x in ganz Österreich

- |   |                                     |                                       |
|---|-------------------------------------|---------------------------------------|
| 1. Singerstraße 14                          | 12. Meidinger Hauptstr. 38          | 3. Landstraßer Hauptstr. 100          |
| 2. Lassallestraße 21                        | 13. Hietzinger Hauptstr. 22         | 15. Sechshauser Straße 33             |
| 3. Landstraßer Hauptstr. 59                 | 14. Hütteldorfer Straße 78          | 15. Sechshauser Straße 46             |
| 5. Reinprechtsdorfer Str. 53                | 15. Mariahilfer Straße 137          | 16. Thaliastraße 37                   |
| 6. Getreidemarkt 1                          | Electronic-Center                   |                                       |
| 6. Mariahilfer Straße 107                   | 17. Hernäuser Hauptstr. 110         | 2340 Mödling, Hauptstr. 2             |
| 7. Burgg. 83, Service Center                | 19. Döblinger Hauptstr. 15 + 17     | 2700 Wr. Neustadt,<br>Stadionstraße 6 |
| 7. Burggasse 94a, KEZ                       | 20. Wallensteinstraße 32            |                                       |
| 7. Kaiserstraße 113-115                     | 21. Brunner Straße 31a              |                                       |
| 9. Alser Straße 38                          | 22. Donauzentrum                    |                                       |
| 9. Franz Josefs Bahnhof<br>(Althanstraße 4) | Donaustadtstraße 1                  |                                       |
| 10. Favontenstraße 76a                      | 2331 Shopping City Süd,<br>Lokal 73 |                                       |
| 11. Simmeringer Hauptstr. 117               | 4820 Bad Ischl, Pfarrgasse 4        |                                       |