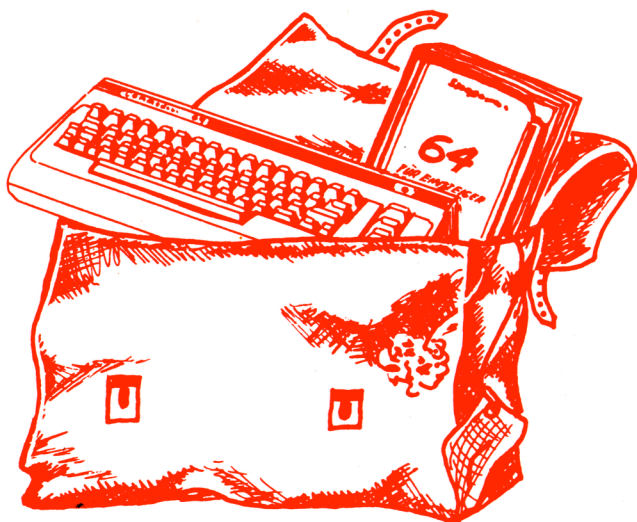


Voß

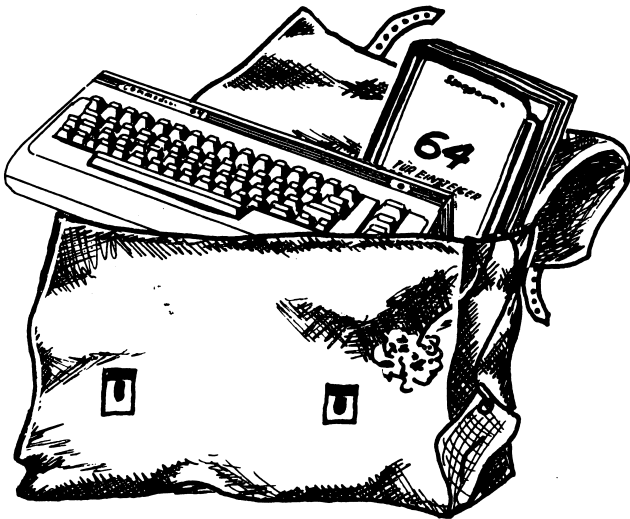
**DAS
SCHULBUCH
ZUM
COMMODORE 64**



EIN DATA BECKER BUCH

Voß

**DAS
SCHULBUCH
ZUM
COMMODORE 64**



EIN DATA BECKER BUCH

ISBN 3-89011-019-3

Copyright (C) 1984 DATA BECKER GmbH
Merowingerstr. 30
4000 Düsseldorf

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form (Druck, Fotokopie oder einem anderen Verfahren) ohne schriftliche Genehmigung der DATA BECKER GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Wichtiger Hinweis!

Die in diesem Buch wiedergegebenen Schaltungen, Verfahren und Programme werden ohne Rücksicht auf die Patentsituation mitgeteilt. Sie sind ausschließlich für Amateur- und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden.

Alle Schaltungen, technische Angaben und Programme in diesem Buch wurden von den Autoren mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. DATA BECKER sieht sich deshalb gezwungen, darauf hinzuweisen, daß weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann. Für die Mitteilung eventueller Fehler sind die Autoren jederzeit dankbar.

VORBEMERKUNG

Über den Einsatz des Computers in der Schule wird sehr viel geredet, aber sehr wenig getan. DATA BECKER hat sich daher entschlossen, selbst ein Buch zum Thema herauszugeben. Geeignet erschien uns dafür der Commodore-64, der unter Schülern meistverbreitete Rechner.

Als Autor konnten wir Werner Voß gewinnen, Professor für Statistik an der Universität Bochum, der nebenberuflich im Bereich der Datenverarbeitung und Programmierung tätig ist. Er hat ein leichtverständliches, interessantes und äußerst vielfältiges Buch geschrieben, das aufgrund seines didaktisch gelungenen Aufbaus nicht nur in Unterrichtsstunden und Seminaren seinen Platz finden wird. So werden sich Computerfreaks plötzlich für Fächer begeistern, die sonst am Rande Ihres Interesses liegen und andererseits werden viele Schüler in das für die Zukunft so wichtige Gebiet der Datenverarbeitung geführt.

Es gibt kaum ein Unterrichtsfach, in dem sich der Commodore-64 nicht gewinnbringend einsetzen läßt. Viel Spaß dabei!



Dr. Achim Becker

GLIEDERUNG

=====

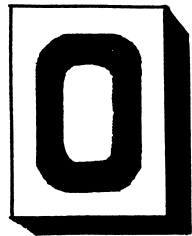
		Seite
∅.	Vorwort	1
1.	Grundelemente der Programmiersprache BASIC	7
1.1	Vorbemerkung	7
1.2	Grundbegriffe, die man kennen sollte	8
1.3	Zur Funktionsweise von Rechnern ..	12
1.4	Problemanalyse	14
1.5	Ergebnisausgabe	17
1.6	Wertzuweisungen	22
1.7	Informationseingabe	26
1.8	Programmverzweigungen	28
1.9	Programmschleifen	32
1.1∅	Die Benutzung externer Speicher ..	34
1.11	Ergänzungen	36
2.	Mathematik	39
2.1	Vorbemerkung	39
2.2	Der Satz des Pythagoras	41
2.3	g.g.T, und k.g.V.	50
2.4	Primzahlenprüfung	59
2.5	Quadratische Gleichung	68
2.6	Euler'sche Zahl	74
2.7	Prozentrechnung	80
2.8	Das Runden von Zahlen	85
2.9	Probleme der Rechengenauigkeit ...	92

	Seite
3. Chemie	95
3.1 Zusätzliche BASIC-Anweisungen .	95
3.2 Das Wassermolekül	103
3.3 Reaktionsgleichung	109
3.4 Stöchiometrisches Rechnen	116
3.5 Das Periodensystem der Elemente	122
4. Physik	131
4.1 Vorbemerkung	131
4.2 Graphik-Programmierung	132
4.3 BASIC-Graphikanweisungen	136
4.4 Die Federwaage	144
4.5 Der Satz des Archimedes	151
4.6 Pendelbewegung	157
4.7 Optische Abbildung	163
4.8 Das Spektrum	168
4.9 Das Ohm'sche Gesetz	173
5. Sprachen	179
5.1 Vorbemerkung und BASIC-Ergänzungen	179
5.2 Englische unregelmäßige Verben	182
5.3 Französisch-Vokabeln	189
5.4 Englisch-Vokabeltest	196
5.5 Das Sortieren von Vokabeln	203
6. Biologie/ökologie	211
6.1 Vorbemerkung	211
6.2 Ungebremstes Wachstum	212
6.3 Gebremstes Wachstum	217

	Seite
6.4 Umweltverschmutzung	223
7. Erdkunde/Geschichte	231
7.1 Vorbemerkung	231
7.2 Historische Jahreszahlen	232
7.3 Die Hauptstädte der Länder	238
7.4 Die Bevölkerungsentwicklung in verschiedenen Nationen	245
8. Wirtschaft	255
8.1 Vorbemerkung	255
8.2 Zinsrechnung	256
8.3 Hypothekentilgung	262
8.4 Arithmetisches Mittel	269
8.5 Häufigkeitsverteilung	273
8.6 Die Wirtschaftskraft der Bundes- länder	282
9. Mathematik II : Hochauflösende Gra- phik	291
9.1 Vorbemerkung	291
9.2 Graphik-Anweisungen	293
9.3 Die Gerade	299
9.4 Der Kreis	307
9.5 Die Sinuslinie	317
10. Ausblick	323
Stichwortverzeichnis	325

Voß C64/ Schule	Kapitel Ø : Vorwort Abschnitt - : -	Seite 1
-----------------------	--	------------

Vorwort
=====



Computer - wie der Commodore C64 - sind nicht nur zum Spielen da. So faszinierend fertige Programme auch sein mögen - Fußballspiele, Weltraumkriege, Autorennen, Schachpartien, Kämpfe mit seltsamen Ungeheuern usw. - noch aufregender ist es, selbst zu programmieren: Wer zum ersten Mal ein Problem so aufbereitet hat, daß das in den Rechner eingegebene Programm dieses Problem in Sekundenschnelle und ohne Fehler tatsächlich löst, der kann sich freuen. Ein solcher Erfolg nämlich zeigt, daß der Computer sinnvolle Arbeit leisten kann, d.h. er kann uns von der Last mühsamer "Kopfarbeit" befreien - und das ist auch sein wesentlicher Verwendungszweck; dies macht die außerordentlich rasch zunehmende Bedeutung dieser Rechner aus.

In diesem Buch soll deshalb gezeigt werden, wie man als Schüler den Rechner, speziell den erfolgreichen Commodore C64, dazu einsetzen kann, wesentliche Probleme aus dem

Voß C64/ Schule	Kapitel Ø : Vorwort Abschnitt - : -	Seite 2
-----------------------	---	------------

Bereich des schulischen Alltags zu lösen.
 Dies bedeutet also, daß mit diesem Buch vier Ziele gleichzeitig verfolgt werden:

1. Typische Schulprobleme beispielsweise aus der Mathematik, aus der Physik und aus anderen Fächern sollen aufgegriffen und die Wege zu ihrer Lösung sollen diskutiert werden.
2. Die Beschreibung der Problemlösungswege soll dem Leser zeigen, wie man ganz allgemein Probleme anpacken muß, damit ein Rechner sie dann für uns lösen kann.
3. Der Leser wird erkennen, wie die Programmiersprache BASIC verwendet werden kann, um diejenigen Computerprogramme zu schreiben, die wir zur Lösung der oben genannten Probleme benötigen.
4. Auf diese Weise erhält zum guten Schluß der Leser eine Sammlung von Programmen, die er nutzbringend im Schulalltag verwenden kann.

Es versteht sich, daß der Nutzen dieses Buchs für den Leser sich insbesondere dann erschließt, wenn er die vorgeführten Programme selbst am Rechner ausprobiert und wenn er auch Alternativprogramme oder Programmveränderungen ausprobiert.

Voß C64/ Schule	Kapitel 0 : Vorwort Abschnitt - : -	Seite 3
-----------------------	--	------------

Im Bereich des Programmierens und des Rechneinsatzes gilt nämlich wie in vielen anderen Lebensbereichen auch:

Übung macht den Meister

Im einzelnen werden in den Kapiteln dieses Buchs die folgenden Problembereiche angesprochen:

1. Probleme aus der Mathematik
2. Probleme aus der Physik
3. Chemische und biologische Fragestellungen
4. Programme für den Fremdsprachenunterricht
5. Probleme aus den Bereichen Geschichte, Erdkunde und Wirtschaftskunde.

Diesen inhaltlichen Kapiteln werden einige Ausführungen zu den Grundlagen und Bedingungen des Rechneinsatzes beigelegt, und wir werden auch die Grundelemente der Programmiersprache BASIC besprechen.

Der Leser, der schon über hinreichende BASIC-Kenntnisse verfügt, mag die entsprechenden Ausführungen getrost überblättern.

Zu den Computerprogrammen, die in den folgenden Kapiteln entwickelt oder vorgestellt werden, ist folgende Anmerkung erforderlich:

Zum einen handelt es sich um Programme, die bestimmte Probleme lösen sollen (Beispiel: Programm zur Ermittlung eines "größten, gemeinsamen Teilers").

Voß C64/ Schule	Kapitel 0 : Vorwort Abschnitt - : -	Seite 4
-----------------------	--	------------

Diese Programme können mit unterschiedlichen Ausgangsdaten "gefüttert" werden und dienen dann immer dem gleichen Zweck. Zum anderen aber gibt es auch reine Trainingsprogramme: Ein Programm zum Beispiel, das zum Erlernen von Englisch-Vokabeln verwendet werden kann, löst keine Probleme, sondern es "trainiert" nur.

Allerdings gilt auch bei diesen Programmen, daß sie allgemeiner eingesetzt werden können, wenn sie mit unterschiedlichem Ausgangsmaterial versorgt werden.

Zu diesem Buch werden auch eine Programmkassette und eine Programm-Diskette vorbereitet, wo sich alle Programme, die hier besprochen werden, wiederfinden, so daß der Leser, der diese Programme benutzen will, sich von der Last des Eintippens befreien kann. Diese Programmsammlung verzichtet allerdings auf die kleinen Beispielprogramme im folgenden Kapitel, die nur dazu dienen, die Grundelemente der Programmiersprache BASIC zu illustrieren.

Die folgende Anmerkung ist für die weiteren Ausführungen besonders wichtig:

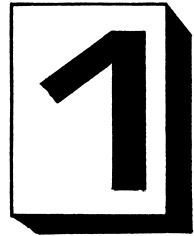
Wir haben in den Programmen, die Schulprobleme lösen sollen, bewußt darauf verzichtet, mit fortgeschritteneren und anspruchsvolleren BASIC-Sprachelementen zu arbeiten. Vielmehr haben wir uns bemüht, bei der Erstellung der Programme nur einfachere und grundlegende Sprachelemente zu benutzen. Dies hat den Zweck, den Leser - auch wenn er noch Anfänger sein sollte - nicht durch programmier-technische Fragen zu überfordern, sondern ihm die Konzentration auf die sachlichen Probleme zu ermöglichen.

Diese Einschränkung hat zur Folge, daß manche Programme etwas unbeholfen wirken mögen - aber sie funktionieren, und zwar mit einfachsten Mitteln; darauf kam es uns zunächst an.

Voß C64/ Schule	Kapitel 0 : Vorwort Abschnitt - : -	Seite 5
<p>Bei komplizierteren Problemen in den späteren Kapiteln wird dann allerdings auch der Einsatz komplizierterer Sprachelemente erforderlich, die dann erst von Fall zu Fall besprochen werden.</p> <p>Weiterhin möge der Leser insbesondere beachten, daß manche der vorgestellten Programme nur exemplarischen Charakter haben:</p> <p>Wenn z.B. in einem Vokabel-Abfrageprogramm nur 10 Vokabeln zur Verfügung stehen, so muß der Benutzer, wenn er dieses Programm konkret nutzen will, zunächst die Vokabelliste verlängern. An der grundsätzlichen Struktur des Programms - und auf die kommt es ja an - ändert sich dadurch natürlich nichts.</p>		

Voß C64/ Schule	Kapitel 0 : Vorwort Abschnitt - : -	Seite 6

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 1 : Vorbemerkung	Seite 7
-----------------------	---	------------



Kapitel 1: Grundelemente der Programmiersprache BASIC

1.1 Vorbemerkung

In diesem Kapitel werden neben einigen allgemeinen Grundbegriffen aus der Datenverarbeitung, die später wieder auftauchen, die wichtigsten Grundelemente der Programmiersprache BASIC dargestellt und kurz erläutert.

Derjenige Leser, der schon hinreichend mit BASIC vertraut ist, kann dieses Kapitel überblättern.

Leser aber, die noch gar keine Programmiererfahrung haben, werden die Ausführungen in diesem Kapitel für etwas zu kurzgefaßt ansehen. Man kann sich aber mit dem Hinweis trösten, daß in den späteren Kapiteln bei der praktischen Verwendung dieser BASIC-Sprachbestandteile deren Verwendung, Funktionsweise und Wirkung dann auf jeden Fall deutlich werden wird.

So gesehen, enthält dieses Kapitel eher nur einen vorläufigen Überblick.

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 2 : Grundbegriffe	Seite 8
-----------------------	--	------------

1.2 Grundbegriffe, die man kennen sollte

Das Wort "Computer" bedeutet wörtlich übersetzt "Rechner". Ein moderner Computer kann aber mehr als nur rechnen. Er kann beispielsweise auch mit Buchstaben, Worten und Texten umgehen, er kann graphische Darstellungen erzeugen, er kann Musik produzieren, er kann Steuerungsaufgaben übernehmen u.ä. Dies bedeutet also, daß die Rechner, die heute manchmal schon für ein paar Hundert Mark zu erwerben sind, nicht nur die Rechenkapazitäten bereitstellen wie z.B. die Großrechner der sechziger Jahre, die damals unerschwinglich teuer waren, sondern sie sind viel universeller einsetzbar.

Daten: Unter "Daten" kann man im weitesten Wortsinn "Informationen" verstehen. Im einzelnen können dies z.B. sein:

- Ziffern und Zahlen
- Werte
- Buchstaben
- Symbole (Sonderzeichen)
- Worte und Texte

Datenverarbeitung: Unter "Datenverarbeitung" versteht man dann alle Prozeduren (häufig rechnerischer Art), derartige Daten zu erfassen, zu speichern, auszuwerten oder zu analysieren und Ergebnisse auszugeben.

Führt man Datenverarbeitung im obigen Sinn nicht per Hand, sondern mit Hilfe von Rechnern (Computern) durch, dann ist es sinnvoll, auch die folgenden Begriffe zu kennen:

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 2 : Grundbegriffe	Seite 9
-----------------------	--	------------

bit: Unter einem bit "binary information digit) versteht man die kleinste Informationseinheit (s.o.: "Daten" = "Informationen").

Die kleinste Informationseinheit nennt man ein bit deshalb, weil in einem bit nur zwei Informationsinhalte vorstellbar sind. Aus der Sicht des Computerherstellers gesprochen, heißt dies: Ein bit kann nur zwei Informationsinhalte speichern.

Diese beiden Informationsinhalte werden üblicherweise bezeichnet mit 0 und 1.

byte: Ein byte ist eine Zusammenfassung einer Folge von bits.
Üblicherweise bilden 8 bits ein byte. Ein byte ist somit in der Lage, eine Folge von 8 Nullen und/oder Einsen zu speichern.

Symbol: Wir unterscheiden drei Gruppen von Symbolen:

1. numerische Symbole = Ziffern
2. alphabetische Symbole = Buchstaben
3. sonstige Symbole = Sonderzeichen

Im allgemeinen wird jedes dieser Symbole im Rechner dargestellt als Folge von 8 Nullen und/oder Einsen, d.h. ein byte (s.o.) nimmt ein Symbol auf.

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 2 : Grundbegriffe	Seite 10
<p><u>Feld:</u> Eine Folge von Symbolen (z.B. Worte oder Zahlen) bilden ein Feld.</p> <p>Beispielsweise besteht das Wort PRINT aus 5 Symbolen, belegt mithin 5 byte im Rechner, und da diese 5 Symbole zusammengehören, bilden sie ein Feld.</p> <p>Entsprechendes gilt für die Zahl 178. Sie wird in einem dreistelligen Feld gespeichert (man sieht, daß es auch einstellige Felder geben kann).</p> <p><u>Variable:</u> Eine Variable ist eine veränderliche Größe oder - präziser gesprochen - ein <u>Name</u>, unter dem verschiedene <u>Werte</u> (s.u.) angesprochen werden können.</p> <p>Beispielsweise stehen hinter dem Variablenamen "Körpergröße" die Werte 178, 185, 167 usw., hinter dem Namen "Kinderzahl" z.B. die Werte 1, 2, 0, 4 usw., hinter dem Namen "Ort" z.B. die Werte "Dortmund", "Bochum", "Kassel" und dergl.</p> <p><u>Wert:</u> Wie aus der vorangegangenen Begriffsklärung schon zu entnehmen ist, verstehen wir unter dem Begriff "Wert" eine bestimmte <u>Ausprägung</u> einer gegebenen Variablen. Es muß sich dabei nicht notwendigerweise um Zahlen handeln, sondern wie das obige Beispiel der Variablen "Ort" zeigt, können als Werte auch Worte oder Texte auftreten.</p> <p>Die Werte belegen jeweils <u>ein</u> Feld im Rechner (s.o.). Man spricht auch von <u>einer Speicherstelle</u>.</p>		

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 2 : Grundbegriffe	Seite 11
	<p><u>String:</u> Einen Text, der z.B. als Variablenausprägung auftreten kann (s.o.), nennt man "string"; allgemein gesprochen handelt es sich um eine Zeichenkette.</p> <p><u>Array:</u> Die Gesamtheit aller Ausprägungen <u>einer</u> Variablen (Zahlen oder Worte) nennt man "array".</p> <p><u>Datensatz:</u> Mehrere zusammengehörige Felder nennt man einen Datensatz. Beispielsweise bilden alle vorhandenen Angaben über eine Person (Name, Geschlecht, Alter, Wohnort etc.) einen Datensatz. In diesem Beispiel wäre ein Datensatz also eine Folge zusammengehöriger Zahlen und/oder Worte. Aber auch eine Folge nur von Worten kann als Datensatz bezeichnet werden, z.B. die <u>Programm-</u> <u>anweisung:</u></p> <p style="text-align: center;">PRINT "OTTO", "SUSI", "OLAF"</p> <p>Der Datensatz wird auch als "record" bezeichnet.</p> <p><u>Datei:</u> Mehrere zusammengehörige Datensätze bilden eine Datei. Beispielsweise erhalten wir eine Datei, wenn wir die Datensätze aller Beschäftigten eines Unternehmens zusammenstellen. Genauso würden wir eine Datei erhalten, wenn wir alle Programm-anweisungen eines Programms insgesamt betrachten.</p> <p>Die Datei wird auch als "file" bezeichnet.</p>	

1.3

Zur Funktionsweise von Rechnern

Es wurde schon erläutert, was unter "Datenverarbeitung" verstanden werden soll. Daraus geht hervor, daß eine Datenverarbeitungsanlage die folgenden Tätigkeiten (Funktionen) ausführen muß:

- | | | | |
|--|---|---|--------------|
| 1. Informationsaufnahme (Dateneingabe) | } | E | Eingabe |
| 2. Informationsspeicherung | | V | Verarbeitung |
| 3. Verarbeiten von Daten im engeren Wortsinn (z.B. Rechnen oder Umstellen von Texten und dergl.) | | A | Ausgabe |
| 4. Ausgabe von Ergebnissen | | | |

Informationseingabe, -verarbeitung und -ausgabe macht eine Reihe organisatorischer Schritte notwendig, die wir hier nicht zu besprechen brauchen, weil sie der Rechner im allgemeinen selbständig erledigt. Verantwortlich dafür ist das sog. Betriebssystem.

Der Rechner muß also von uns Informationen erhalten, damit ein Datenverarbeitungsprozeß in Gang kommen kann. Er gibt seinerseits Informationen zurück, beispielsweise errechnete Ergebnisse.

Diejenigen Informationen, die wir ihm geben müssen, lassen sich in drei Informationsgruppen einteilen:

1. Zu verarbeitende Daten
2. Programmanweisungen
(Arbeitsschritte, die notwendig sind, um die eingegebenen Daten den gewünschten Auswertungsprozeduren zu unterziehen)
3. Kommandos
(Mitteilungen und Anweisungen an das Betriebssystem des Rechners; s.O.)

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 3 : Funktionsweise	Seite 13
<p>Entsprechend den unterschiedlichen Teilaufgaben, die der Rechner erledigen muß, unterteilen wir:</p> <ol style="list-style-type: none"> 1. Eingabegeräte (beim C64 die Tastatur) 2. Verarbeitungsbereich (die sog. Zentraleinheit) 3. Ausgabegeräte (Bildschirm oder z.B. Drucker) <p>Darüber hinaus gibt es sog. <u>externe</u> (oder periphere) Speicher. Sie dienen dazu, Programme oder Datenbestände auf Dauer aufzubewahren. In der Regel handelt es sich dabei um Tonband-<u>Kassetten</u> oder um sog. <u>Disketten</u>, die in entsprechenden Laufwerken bespielt bzw. gelesen werden können.</p>		

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 4 : Problemanalyse	Seite 14
-----------------------	---	-------------

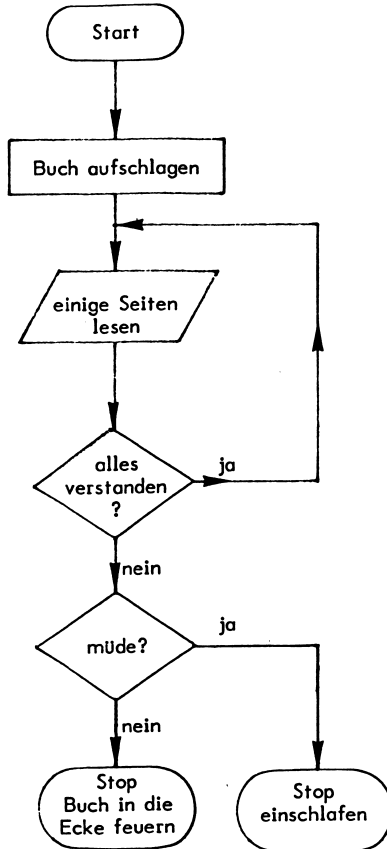
1.4

Problemanalyse

Wenn ein bestimmtes Problem mit Hilfe eines Computers gelöst werden soll, so ist dies nur dann möglich (der Leser möge diese Anmerkung immer beachten), wenn der Computerbenutzer den Lösungsweg vorher schon gefunden hat. Das heißt, daß die Problemlösung an sich, trotz aller modernen Rechner, immer noch nach wie vor nur im Kopf und nicht in der Maschine erledigt werden kann; der Rechner entlastet uns dann nur von der Aufgabe, den Lösungsweg Schritt für Schritt abzuarbeiten.

Damit ein Rechner nun diese Abarbeitung leisten kann, muß ihm der eigentliche Lösungsweg als Folge einzelner Anweisungen Schritt für Schritt vorgezeichnet werden. Dies wiederum macht es erforderlich, daß ein zu lösendes Problem zunächst gedanklich in eine logische Folge von Einzelschritten zerlegt wird - wir bezeichnen diesen Vorgang als Problemanalyse.

Häufig ist es bei einer derartigen Problemanalyse hilfreich, wenn man die Folge von Arbeitsschritten graphisch in Form eines sog. Flußdiagramms darstellt. Dies zeigt anschaulich das folgende Beispiel:



Quelle : W.VOSS : BASIC leicht und schnell gelernt
am alphaTronic PC, Darmstadt 1984,
(heim-Verlag), Seite 8.

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 4 : Problemanalyse	Seite 16
-----------------------	---	-------------

In den folgenden Kapiteln werden wir auf dieses Instrument der "Problemzergliederung" zurückgreifen, weil auf diese Weise das Programmieren, d.h. die Erstellung der Anweisungsfolge für den Computer, wesentlich erleichtert wird und logische Programmfehler eher vermieden werden können.

1.5 Ergebnisausgabe

Eine Folge von Anweisungen, die einen Rechner veranlassen, ein vorgegebenes Problem selbständig zu lösen, nennt man ein Programm. Es muß in einer Sprache geschrieben werden, die der Rechner verstehen kann. Eine solche Sprache ist die Programmiersprache BASIC.

```
BASIC =  
    Beginner's  
    All-Purpose  
    Symbolic  
    Instruction  
    Code
```

Jedes BASIC-Programm besteht aus einer Folge von Anweisungen (Statements), die dem Rechner zeilenweise eingegeben werden. Eine Zeile nennen wir Satz.

Regel 1: Jeder Satz in BASIC benötigt eine Satznummer

Regel 2: Jeder Satz muß mit der RETURN-Taste abgeschlossen werden.

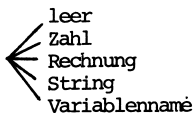
Regel 3: In einem Satz können mehrere Statements stehen, die dann durch Doppelpunkte voneinander zu trennen sind.

Statement 1:

nn END

Das END-Statement beschließt das BASIC-Programm.

Statement 2:

nn PRINT 

Das PRINT-Statement bringt Ergebnisse auf den Bildschirm. Man unterscheidet fünf Möglichkeiten:

1. leer : Ergibt eine Leerzeile
2. Zahl : Die Zahl wird ausgegeben
3. Rechnung : Das Ergebnis der Rechnung wird ausgegeben
4. String : Der String wird ausgegeben (ein String ist eine Zeichenkette, die in Anführungszeichen einzuschließen ist)
5. Variablenname : Der Inhalt des Speicherfeldes mit diesem Namen wird ausgegeben

Das Wort PRINT darf durch ein ? abgekürzt werden.

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 5 : Ergebnisausgabe	Seite 19
-----------------------	--	-------------

Das folgende einfache BASIC-Programm benutzt alle diese Möglichkeiten :

```

1Ø PRINT 3
2Ø PRINT
3Ø PRINT 4.8/2
4Ø ?
5Ø ?"SUSI"
6Ø ?
7Ø ? X
8Ø END

```

Ein solches BASIC-Programm wird erst abgearbeitet, wenn wir das Kommando RUN eingeben:

```

Kommando 1:
      RUN

```

Für das obige Beispiel erhalten wir nach dem Kommando RUN auf dem Bildschirm die folgenden Ergebnisse:

```

3
2.4
SUSI
Ø

```

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 5 : Ergebnisausgabe	Seite 20
-----------------------	--	-------------

Regel 4: Kommandos erhalten keine Satznummer

Wollen wir nach dem Ergebnisausdruck unser Programm auf dem Bildschirm noch einmal sehen, benötigen wir das folgende Kommando:

Kommando 2:

LIST

Zum PRINT-Statement sind noch einige Anmerkungen erforderlich:

Die verschiedenen Möglichkeiten, die in Statement 2 genannt wurden, können in einem PRINT-Statement mehrfach und auch gemischt verwendet werden.

Sie sind dann durch Kommata oder durch Strichpunkte voneinander zu trennen.

Regel 5: Werden die verschiedenen Ausdrücke im PRINT-Statement durch Kommata getrennt, wird der Ausdruck zu Beginn der nächsten Bildschirmzone fortgesetzt; wird das Semikolon als Trenner benutzt, geht es unmittelbar weiter.

Regel 6: Jeder PRINT-Befehl bewirkt in der Regel auch einen Zeilenvorschub. Steht im Statement nur das Wort PRINT, so erfolgt nur ein Zeilenvorschub.

Regel 7: Ein Komma oder ein Semikolon am Ende eines PRINT-Statements unterdrückt den Zeilenvorschub.

Das PRINT-Statement erzeugt Ergebnisausdrücke, die am linken Bildschirmrand beginnen. Will man dies ändern, kann man sich der TAB-Funktion (TAB = Tabulator) bedienen.

PRINT TAB(10) 8

rückt den Ausdruck des Wertes 8 zehn Stellen nach rechts.

Auf den Begriff des Variablennamens, der im PRINT-Statement ja auch schon eine Rolle spielte und auf Einzelheiten der ausführbaren Rechnungen, kommen wir im folgenden Abschnitt zu sprechen.

Will man vor der Eingabe eines neuen Programms ein eventuell schon vorhandenes, altes und nun nicht mehr benötigtes Programm löschen, benötigt man das folgende Kommando, welches den Inhalt des Programmspeichers löscht:

Kommando 3:

NEW

1.6

Wertzuweisungen

Für viele Zwecke ist es - wie man noch sehen wird - sehr sinnvoll, wenn Speicherplätze im Rechner mit Namen versehen werden können, damit dann unter diesen Namen unterschiedliche Werte gespeichert werden können.

Zunächst genügt es, drei Typen von Variablen zu unterscheiden:

1. Reelle Variablen
2. Ganzzahlige Variablen
3. Stringvariablen

Reelle Variablen können als Werte reelle Zahlen annehmen, also z.B.: 3.5 -17.Ø1 18. Ø.Ø4 -7.5 etc. (statt des Dezimalkommata wird beim Programmieren ein Punkt verwendet)

Ganzzahlige Variablen können als Werte nur ganze Zahlen annehmen, also z.B.: 4 8 -3 Ø 2Ø etc.

String-Variablen haben als Werte Symbolfolgen, die in Anführungszeichen einzuschließen sind, also z.B.:

"OTTO" "ØØ" "DORTMUND" "A-15" usw.

Regel 8: Die Namen reeller Variablen bestehen aus einem oder zwei alphanumerischen Zeichen, wobei das erste ein Buchstabe sein muß. Die Namen ganzzahliger Variablen werden zusätzlich mit % versehen. Die Namen von Stringvariablen werden stattdessen zusätzlich mit \$ versehen.

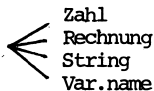
Bei vielen Rechnern dürfen die Namen auch länger sein (Benutzerhandbücher konsultieren!).

Beispiele:

1. Reelle Variablen: X B1 NN
2. Ganzzahlige Variablen: C% Y1% AA%
3. Stringvariablen: F\$ C2\$ ZZ\$

Nach dieser Begriffsklärung können wir das LET-Statement besprechen, welches der Wertzuweisung dient, so daß für einen gesamten Programmablauf dieser Wert dann zur Verfügung steht:

Statement 3:

nn	LET	Var.name	=	
----	-----	----------	---	--

Das Wort LET darf auch entfallen.

Mit diesem Statement kann also, wie auch übrigens schon mit dem PRINT-Statement, gerechnet werden.

Die Rechenoperatoren sind die folgenden:

- + Addition
- Subtraktion
- * Multiplikation
- / Division
- ^ Potenzierung

Die Rangfolge der Operatoren entspricht der aus der Schulmathematik bekannten; bei Bedarf kann durch Klammersetzung davon abgewichen werden.

Mit dem LET-Statement können also auch schon kompliziertere Anweisungen geschrieben werden. Wenn man sich dabei vertippt, besteht die einfachste Korrektur darin, den fehlerhaften Satz per RETURN-Taste abzuschließen und ihn darunter einfach neu zu schreiben. Soll ein Satz ganz gelöscht werden, so gibt man einfach seine Satznummer ein plus RETURN-Taste.

Soll ein Satz nachträglich in ein Programm eingefügt werden, so wird dies durch Vergabe passender "Zwischen-Satznummern" erreicht.

Viele Rechnungen können durch die sog. eingebauten Funktionen erleichtert werden. Die wichtigsten dieser Funktionen sind die folgenden:

Allgemeine Schreibweise:

Var.name = Funktionsname (Argument)

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 6 : Wertzuweisungen	Seite 25
-----------------------	--	-------------

Wichtige Funktionen

<u>Name</u>	<u>Aufgabe</u>
SIN COS TAN	berechnet die Winkelfunktionswerte, wobei als Argument Einheiten des Kreisparameters zu verwenden ist
LOG EXP	natürlicher Logarithmus; Umkehrung dazu
SQR	Quadratwurzel
ABS	Absolutwert
INT	größter, in einem reellen Wert vor- handener, ganzzahliger Teil
RND	Zufallszahlenerzeugung
CHR\$	Ausgabe eines ASCII-Code-Zeichens

Beispiele zu diesen Funktionen sind hier entbehrlich.
Sie finden sich zur Genüge in den folgenden Kapiteln.

Nur die letzte Funktion soll in der Anwendung, in
der sie uns immer wieder begegnen wird, kurz vorge-
stellt werden:

```
PRINT CHR$(147)
```

"räumt" den Bildschirm.

Vor jeder Ergebnisausgabe ist dies eine sehr nützliche
Anweisung und wird deshalb von uns regelmäßig benutzt.

1.7

Informationseingabe

Zur Informationseingabe eignet sich hervorragend das folgende Statement:

Statement 4:

```
nn INPUT ("Text";) Var.liste
```

Dieses Statement sieht recht kompliziert aus, ist aber leicht zu verstehen:

Gelangt der Rechner an dieses Statement, unterbricht er die Programmabarbeitung und produziert ein Fragezeichen auf dem Bildschirm (zusätzlich zu dem eventuell angegebenen "Text", der aber auch weggelassen werden kann; deshalb oben in Klammern). Er erwartet dann vom Benutzer so viele Werte (durch Kommata zu trennen), wie Variablennamen in der Variablenliste genannt sind (also mindestens einen Wert, wenn nur ein Name genannt ist).

Beispiele:

```
INPUT X      Der Rechner erwartet eine Zahl,  
             die dann im Feld X gespeichert  
             wird.
```

```
INPUT A,B,C  Der Rechner erwartet drei Zahlen.
```

INPUT "BITTE NAMEN EINGEBEN";N\$:

Der Rechner druckt den String,
der in Anführungszeichen steht,
und erwartet dann einen String
(z.B. "OTTO"), der im Feld N\$
gespeichert wird.

Regel 9: Die Variablennamen in der Variablen-
liste des INPUT-Statements werden
durch Kommata getrennt.
Die einzugebenden Werte müssen genauso
getrennt werden.

Beispiel :

```
1Ø INPUT "BITTE 4 WERTE : ";A,B,C,D
2Ø LET S = A + B + C + D
3Ø LET AM = S/4
4Ø PRINT"MITTELWERT = ";AM
5Ø END
```

Dieses Programm berechnet offenbar den Durchschnitt
aus beliebigen vier einzugebenden Zahlen.

1.8

Programmverzweigungen

Solange in einem BASIC-Programm keine Verzweigungen auftreten (sog. Sprünge), erfolgt die Abarbeitung in der Reihenfolge der vergebenen Satznummern.

Will man von dieser Reihenfolge abweichen, so sind Programmsprünge erforderlich.

Wir unterscheiden unbedingte Sprünge von bedingten Sprüngen.

Betrachten wir zunächst den bedingten Sprung, der mit Hilfe des folgenden Statements möglich ist:

Statement 5:

```
nn  IF  Bedingung THEN < Anweisung  
                                Satznummer mm
```

Dieses Statement veranlaßt das Betriebssystem des Rechners zu einer Abfrage: Wird die Bedingung, die nach dem Befehlswort IF steht, erfüllt, so wird der hinter dem THEN stehende Befehl ausgeführt bzw. es wird zum Satz mit der Nummer mm verzweigt.

Ist die Bedingung hingegen nicht erfüllt, so geht die Programmsteuerung zu dem Satz über, der nach dem IF-Statement folgt.

Regel 10:

Ist die Bedingung im IF-Statement erfüllt, so wird die Anweisung hinter THEN ausgeführt bzw. zu der dort stehenden Satznummer verzweigt.

Ist die Bedingung nicht erfüllt, wird der nächste Satz bearbeitet.

Regel 11:

Stehen hinter dem IF-Statement noch weitere Statements im gleichen Satz, so werden diese nur dann ausgeführt, wenn die Bedingung erfüllt ist.

Beispiel :

```
1Ø I = 1
2Ø Q = I*I
3Ø W = SQR(I)
4Ø PRINT I,Q,W
5Ø I = I + 1
6Ø IF I<=2Ø THEN 2Ø
7Ø END
```

Dieses Programm bestimmt offensichtlich für alle ganzen Zahlen von 1 bis 2Ø (jeweils im Feld I zu finden), die Quadratzahlen (im Feld Q; Satz 2Ø) und die Quadratwurzeln (im Feld W; Satz 3Ø) und druckt die jeweils zusammengehörigen drei Werte zeilenweise auf dem Bildschirm.

Entscheidend dabei ist Satz 6Ø. Er besagt, daß das Programm immer wieder zum Satz 2Ø zurückkehren soll, solange im Feld I nach der Erhöhung des Inhalts dieses Feldes jeweils um 1 (Satz 5Ø) ein Wert steht, der kleiner als 21 ist.

Solange diese Bedingung erfüllt ist, wird das getan, was hinter dem THEN steht (nämlich Rücksprung nach Satz 2Ø); ist sie nicht mehr erfüllt, wird der dem IF-Statement folgende Satz (Satz 7Ø) bearbeitet.

Die Alternative zum bedingten Sprung ist der unbedingte Sprung:

Statement 6:

```
nn GOTO mm
```

Erreicht die Programmsteuerung dieses Statement, so erfolgt ein Sprung zum Satz mit der Nummer mm.

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 8 : Programmverzweigungen	Seite 31
-----------------------	--	-------------

```

1Ø ?CHR$(147)
2Ø INPUT"BITTE WERT EINGEBEN : ";X
3Ø N=N+1
4Ø S=S+X
5Ø INPUT"NOCH EIN WERT (J/N) : ";A$
6Ø IF A$ = "J" THEN 2Ø
7Ø AM = S/N
8Ø ?"MITTELWERT = ";AM
9Ø END

```

Dieses Programm bestimmt für eine beliebige Zahl einzugebender Werte das arithmetische Mittel. Wenn ein Wert gegeben wurde, fragt der Rechner den Benutzer, ob noch einer einzugeben ist. Antwortet der Benutzer mit "J" (für "JA"), so erfolgt ein Sprung von Satz 6Ø zu Satz 2Ø und der nächste Wert wird angefordert.

Ist hingegen kein weiterer Wert vorhanden, d.h. antwortet der Benutzer beim INPUT-Statement in Satz 5Ø nicht mit "J", so erfolgt die weitere Programmabarbeitung ab Satz 7Ø (Berechnung und Ausgabe des Mittelwerts).

1.9

Programmschleifen

Häufig ist es sinnvoll, daß bestimmte Programmteile mehrfach zu durchlaufen sind. Zu diesem Zweck benötigt man sog. Schleifen (loops). Solche Schleifen können mit denjenigen Statements, die wir bisher besprochen haben, leicht erzeugt werden, wie das Beispiel des vorangegangenen Programms zeigt.

Mit den folgenden Statements geht die Schleifenkonstruktion einfacher und es werden zudem zusätzliche Möglichkeiten eröffnet.

```
Statement 7:  nn FOR  Laufvarname = Anfangswert  
                TO    Endwert (STEP Schrittweite)
```

Das folgende Statement gehört notwendig dazu:

```
Statement 8:  nn NEXT Laufvarname
```

("Laufvarname" steht dabei als Abkürzung für :
"Name einer Laufvariablen").

Die Laufvariable, die in diesen beiden Statements auftaucht, muß eine reelle Variable sein. Anfangswert, Endwert und Schrittweite können Zahlen, Variablen oder arithmetische Ausdrücke sein.

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 9 : Programmschleifen	Seite 33
-----------------------	--	-------------

"Gelangt der Rechner an das FOR-Statement, so wird zunächst die Laufvariable gleich dem Anfangswert gesetzt und der Programmteil zwischen FOR und NEXT wird mit diesem Wert durchlaufen.

Bei Erreichen des NEXT wird nun die Laufvariable auf den nächsten Wert, nämlich Anfangswert + Schrittweite, gesetzt und wieder derselbe Programmteil abgearbeitet, jetzt mit dem 2. Wert der Laufvariablen.

Dies erfolgt solange, bis bei Erreichen des NEXT durch weitere Addition der Schrittweite der Endwert überschritten wird. Dann wird der nach NEXT folgende Satz bearbeitet.

"STEP Schrittweite" kann aus dem FOR-Befehl weggelassen werden, wenn die Schrittweite gleich eins sein soll. " (PRUST, 1982, S. 64).

Das folgende Programm druckt 10 mal "GUTEN TAG" untereinander:

```

10 FOR I = 1 TO 10
20 ?"GUTEN TAG"
30 NEXT I
40 ?
50 ?"ENDE"
60 END


```

1.10 Die Benutzung externer Speicher

Gelungene Programme möchte man gern auf Dauer aufheben, so daß man bei Bedarf wieder darauf zurückgreifen kann (entsprechend gilt, daß man ab und zu gern schon vorhandene Programme anderer Verfasser benutzen möchte). Dies wird ermöglicht, wenn man Tonbandkassetten oder Disketten benutzt. Dabei muß man die folgenden Kommandos kennen.

Zum Laden eines Programms von einer Kassette benötigen wir:

Kommando 4:
LOAD "Name des Programms"

Meldet der Rechner, daß er das gesuchte Programm gefunden hat, so wird es nach Drücken der -Taste geladen.

Zum Laden von einer Diskette genügt das folgende Kommando:

LOAD "Name des Programms", 8

Zum Speichern eines Programms auf Kassette benötigen wir:

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 10 : Benutzung externer Speicher	Seite 35
-----------------------	---	-------------

Kommando 5:

SAVE "Name des Programms"

Das entsprechende Kommando für die Diskette lautet:

SAVE "Name des Programms", 8

Sollen "fabrikneue" Disketten zum Speichern von Programmen verwendet werden, so müssen diese zunächst formatiert werden (man spricht auch von "Initialisieren"). Dies kann mit folgender Anweisung, die hier nicht erläutert werden soll (vergl. dazu die Bedienungshandbücher), geschehen:

OPEN 1,8,15

PRINT #1, "N: Name der Diskette, nn"

(siehe VC 1541 Floppy Disk, Bedienungshandbuch, S. 11/12)

Will man feststellen, welche Programme sich alle auf einer Diskette befinden, will man sich also ein Inhaltsverzeichnis ausgeben lassen, so ist einzugeben:

LOAD "\$",8

und, wenn das Diskettenlaufwerk wieder steht:

LIST

1.11

Ergänzungen

Ein paar wenige, nützliche Ergänzungen mögen dieses zusammenfassende Übersichtskapitel beschließen.

Es sieht ganz gut aus, wenn der Bildschirm bei der Programmabarbeitung zunächst gelöscht wird, bevor die ersten Ergebnisse ausgegeben werden. Dies wird ermöglicht mit der schon erwähnten Anweisung :

```
nn PRINT CHR$(147)
```

Häufig ist es sinnvoll, einen Programmablauf zu stoppen, beispielsweise dann, wenn man eine sog. Endlosschleife programmiert hat. Ermöglicht wird dies durch Benutzung der RUN/STOP -Taste.

Schließlich ist anzumerken, daß mit dem Statement

```
Statement 9:      nn      STOP
```

der Programmablauf vom Programm selbst unterbrochen werden kann. Soll das unterbrochene Programm weiter abgearbeitet werden, benötigen wir das Kommando:

```
Kommando 6:      CONT
```

Voß C64/ Schule	Kapitel 1 : BASIC Abschnitt 11 : Ergänzungen	Seite 37
-----------------------	---	-------------

Schließlich sei ergänzend angemerkt, daß es häufig sehr nützlich ist, in ein Programm erklärende Anmerkungen (Überschriften, Kommentare, Erläuterungen usw.) einzufügen. Dies kann mit dem folgenden Statement geschehen:

<u>Statement 10</u> :	nn	REM	Text
-----------------------	----	-----	------

Dieses Statement hat keinen Einfluß auf die Programmabarbeitung, erscheint aber in jeder Programmliste, die erstellt wird.

Diese Ausführungen mögen genügen, um auch dem Anfänger eine Vorstellung von den Grundelementen der Programmiersprache BASIC zu vermitteln. So wird er in die Lage versetzt, die Bausteine der folgenden Anwendungsprogramme zu verstehen. Sollten dennoch Verständnisprobleme bleiben, so gilt der folgende Ratschlag:

A U S P R O B I E R E N

Beim Ausprobieren von Programmen und beim Versuch, gedanklich nachzuvollziehen, warum der Rechner genau das tut, was er tut, lernt man am meisten.

VoB C64/ Schule	Kapitel 1 : BASIC Abschnitt 11 : Ergänzungen	Seite 38

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 1 : Vorbemerkung	Seite 39
-----------------------	--	-------------



Kapitel 2 : Mathematik
=====

2.1 Vorbemerkung

Die Mathematik ist derjenige Bereich, der uns am ehesten einfällt, wenn die Frage gestellt wird, wofür denn überhaupt Computer eingesetzt werden. Deshalb wollen wir mit diesem Bereich die inhaltlichen Ausführungen beginnen. Wir werden später aber sehen, daß die Mathematik nur ein Anwendungsbereich unter vielen ist.

Allerdings stehen wir nun, wie auch in den folgenden Kapiteln, vor dem Problem, welcher Art die Beispiele sein sollen, die hier zweckmäßigerweise aufzunehmen sind. Die Mathematik umfaßt ja bekanntlich ein sehr weites Spektrum unterschiedlicher Schwierigkeitsgrade - beginnend vielleicht bei den Grundrechenarten und bei der Differential- und der Integralrechnung noch lange nicht endend.

Wir wollen das hier so handhaben, daß wir Probleme "mittleren Schwierigkeitsgrades" aufgreifen, also uns etwa an diejenigen Aufgabenstellungen orientieren, wie sie zum Beispiel in der gymnasialen Mittelstufe, also in den

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 1 : Vorbemerkung	Seite 40
-----------------------	--	-------------

Jahrgangsstufen 8 bis 10 auftreten. Sie eignen sich sehr gut dazu, die Möglichkeiten von Rechnern zu demonstrieren, ohne daß wegen des eventuell zu hohen Komplexitätsgrades die Darstellung der behandelten Probleme zu viel Platz in Anspruch nehmen würde.

Absicht dieses Buches ist es ja nicht, den Leser durch die Anwendungsbeispiele zu überfordern, sondern er soll erkennen, was Rechner leisten. Dies ist auch mit einfach strukturierten Fragestellungen möglich, die dem Schulalltag auch der jüngeren Leser angepaßt sind.

Für die "Spezialisten", die sich mit sehr anspruchsvollen mathematischen Problemen beschäftigen wollen, wird ein Fortsetzungsband zu diesem Buch vorbereitet, in dem BASIC-Programme nur für mathematische Probleme vorgestellt werden.

Für die Programme in diesem und in den folgenden Kapiteln gilt die folgende wichtige Anmerkung :

Bei den Programmen wird darauf verzichtet, Eingabefehler der Benutzer "abzufangen"!

Wenn also zum Beispiel bei einem bestimmten mathematischen Problem nur positive Zahlen verwendet werden können (so etwa beim Wurzelziehen), so ist im entsprechenden Programm nicht vorgesehen, dem Benutzer dann zu informieren, wenn er irrtümlich eine negative Zahl eingegeben hat.

Mit einem sehr einfachen Beispiel "mittleren Schwierigkeitsgrades" wollen wir nun beginnen.

2.2 Der Satz des Pythagoras

Am berühmten Lehrsatz des Pythagoras kann in anschaulicher Weise gezeigt werden, wie die einzelnen Beispiele im folgenden gegliedert werden.

Wir werden jeweils die folgenden Arbeitsschritte vollziehen :

Arbeitsschritte :

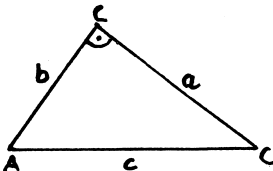
1. SCHRITT : Vorstellung des Problems
2. SCHRITT : Problemanalyse
3. SCHRITT : Flußdiagramm
4. SCHRITT : Programm
5. SCHRITT : Variablenliste
6. SCHRITT : Programmbeschreibung
7. SCHRITT : Ergebnisse

Wenden wir uns nun, dieser Einteilung entsprechend, dem hier genannten Problem zu :

1. SCHRITT : Vorstellung des Problems

Der Lehrsatz des Pythagoras besagt, daß im rechtwinkligen Dreieck die Länge der Grundseite (Hypothese) sich ergibt als Quadratwurzel aus der Summe der beiden Kathetenquadrate.

Dies illustriert die folgende Abbildung :



$$c = \sqrt{a^2 + b^2}$$

2. SCHRITT : Problemanalyse

Bei diesem Problem ist die Problemanalyse sehr einfach :

Man muß sich dabei, wie auch bei allen folgenden Problemen auch, daran erinnern, daß ganz allgemein ein Datenverarbeitungsprozeß nach dem schon vorgestellten

E V A - Prinzip

abläuft.

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 2 : Pythagoras	Seite 43
-----------------------	--	-------------

E	=	Eingabe
V	=	Verarbeitung
A	=	Ausgabe

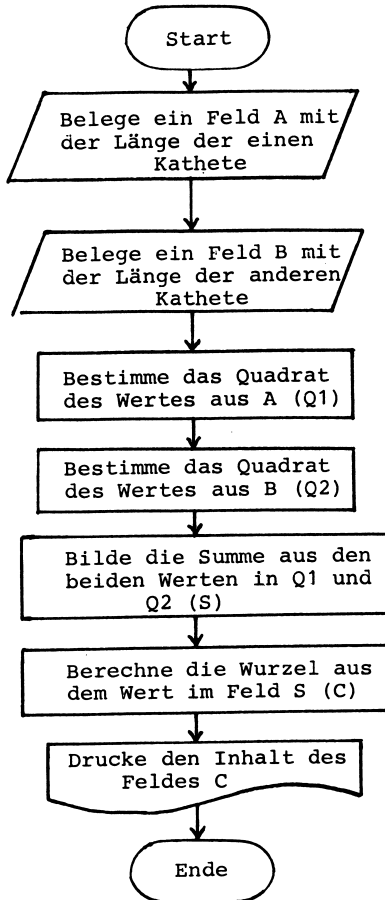
Wir müssen bei der Problemanalyse also zunächst festlegen, welche Informationen der Rechner von uns bekommen muß, damit das gestellte Problem gelöst werden kann.

Die notwendigen Eingabeinformationen sind offensichtlich die Längen der beiden Kathetenseiten a und b.

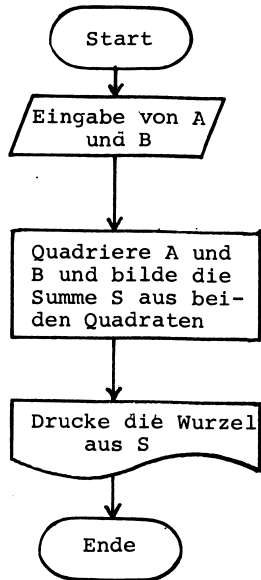
Im Verarbeitungsschritt sind diese beiden Werte zu quadrieren; es ist die Summe dieser beiden Quadrate zu bilden und aus dieser Summe muß dann die Quadratwurzel gezogen werden.

Der Ausgabeschritt besteht lediglich darin, das errechnete Ergebnis (die Länge der Hypotenuse c) auf dem Bildschirm auszugeben.

3. SCHRITT : Flußdiagramm



Kürzere Version des Flußdiagramms :



Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 2 : Pythagoras	Seite 46
-----------------------	--	-------------

4. SCHRITT : Programm

```

10 REM M1-PYTHAGORAS
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR BESTIMMUNG DER HYPOTHENUSE"
40 PRINT"      EINES RECHTWINKLIGEN DREIECKS."
50 PRINT:PRINT:PRINTTAB(10)"PROF.DR.W.VOSS, 1984":PRINT
60 INPUT"ERSTE KATHETE   : ";A
70 PRINT:INPUT"ZWEITE KATHETE   : ";B
80 Q1=A*A
90 Q2=B*B
100 S=Q1+Q2
110 C=SQR(S)
120 PRINT:PRINT:PRINT"HYPOTHENUSE C = ";C
130 END

```

Kürzere Programmversion

```

10 bis 50 : wie oben
60 INPUT "ZWEI KATHETENWERTE : ";A,B
70 C=SQR(A*A+B*B)
80 PRINT:PRINT "HYPOTHENUSE C = ";C:END

```

5. SCHRITT : Variablenliste

```

A : Erste Kathete
B : Zweite Kathete
C : Hypothenuse
Q1 : Quadrat der ersten Kathete A
Q2 : Quadrat der zweiten Kathete B
S : Summe der Kathetenquadrate

```

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 2 : Pythagoras	Seite 47
-----------------------	--	-------------

⑥ SCHRITT : Programmbeschreibung

Bei der Beschreibung beziehen wir uns auf die erste, die längere Programmversion :

Satz 1 ϕ : Kommentar

Satz 2 ϕ : Der Bildschirm wird freigemacht

Satz 3 ϕ -5 ϕ : Titelausdruck auf dem Bildschirm

Satz 6 ϕ : Für die erste Dreiecks-Kathete wird ein Wert angefordert und im Feld A gespeichert

Satz 7 ϕ : Entsprechend zweite Kathete für Feld B

Satz 8 ϕ : Berechnung des ersten Kathetenquadrats

Satz 9 ϕ : Berechnung des zweiten Kathetenquadrats

Satz 10 ϕ : Bestimmung der Summe der beiden Kathetenquadrate

Satz 11 ϕ : Bestimmung der Wurzel aus der Quadratsumme

Satz 12 ϕ : Ausdruck des Ergebnisses auf dem Bildschirm

Satz 13 ϕ : Beendigung des Programms

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 2 : Pythagoras	Seite 48
-----------------------	--	-------------

⑦ SCHRITT : Ergebnisse

Nach dem Starten des Programms (erste Programmversion) erscheint auf dem Bildschirm die Überschrift :

PROGRAMM ZUR BESTIMMUNG DER HYPOTHENUSE
EINES RECHTWINKLIGEN DREIECKS.

PROF.DR.W.VOSS, 1984

Zwei Zeilen tiefer erhalten wir die erste Anforderung :

ERSTE KATHETE : ?

Geben wir daraufhin zum Beispiel den Wert 3 ein, so reagiert der Rechner mit :

ZWEITE KATHETE : ?

Geben wir beispielsweise den Wert 4 ein, so antwortet der Rechner mit :

HYPOTHENUSE C = 5

Dieses Programm eignet sich also dazu, für beliebige Eingaben die Hypothense rechtwinkliger Dreiecke auszurechnen. In diesem Sinn ist es also ganz allgemein gehalten.

VoB C64/ Schule	Kapitel 2 : Mathematik Abschnitt 2 : Pythagoras	Seite 49
<p data-bbox="200 352 954 544">Wir haben dieses Problem hier sehr ausführlich behandelt, um die Vorgehensweise bei den einzelnen Beispielen im Detail zu illustrieren. Sicherlich werden wir uns bei den folgenden Beispielen daraufhin kürzer fassen können, zu mindest wenn sie so einfach sind wie der Lehrsatz des Pythagoras.</p>		

2.3 g.g.T. und k.g.V.

1. SCHRITT : Vorstellung des Problems

Für viele algebraische Rechnungen ist es, insbesondere beim Auftreten von Brüchen, sinnvoll, den größten gemeinsamen Teiler (g.g.T.) und das kleinste gemeinsame Vielfache (k.g.V.) verschiedener Zahlen zu kennen.

Beim g.g.T. handelt es sich um den Wert, durch den zwei Zahlen ohne Rest dividiert werden können, ohne daß es eine größere Zahl gibt, für die dies möglich wäre.

Beim k.g.V. handelt es sich um die Zahl, die durch zwei Ausgangszahlen ohne Rest dividiert werden kann, ohne daß es eine kleinere Zahl gibt, für die dies gilt.

Haben wir zum Beispiel die beiden Zahlen $Z_1 = 3\phi$ und $Z_2 = 4\phi$, so ist der g.g.T. offenbar 1ϕ und das k.g.V. hat den Wert 12ϕ . (Der Leser möge dies durch Kopfrechnen überprüfen).

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 3 : g.g.T. und k.g.V.	Seite 51
-----------------------	---	-------------

②. SCHRITT : Problemanalyse

Wie findet man nun bei zwei gegebenen Zahlen den g.g.T. und das k.g.V.?

Prüfen wir einmal anhand der beiden Zahlen 3ϕ und 4ϕ , wie sinnvollerweise vorzugehen ist, wenn man den g.g.T. finden will :

1. Schritt : $4\phi : 3\phi = 1 \text{ Rest } 1\phi$

2. Schritt : $3\phi : 1\phi = 3 \text{ Rest } \phi$

In diesem Beispiel finden wir also den g.g.T. folgendermaßen :

1. Schritt : Division der größeren durch die kleinere Zahl; Notieren des Rests.

2. Schritt : Division der kleineren Zahl durch den Rest; Notieren des neuen Rests.

Wird dabei der neue Rest null, so ist der Divisor, der zu diesem Nullrest führte, der gesuchte g.g.T.

Dazu noch ein zweites Beispiel zur Verdeutlichung :

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 3 : g.g.T. und k.g.V.	Seite 52
-----------------------	---	-------------

Feld1 : Z1 (größere Zahl)	Feld2 : Z2 (kleinere Zahl)	Division	Rest
78	42	1	36
42	36	1	6
36	6	6	∅

Der g.g.T. von 78 und 42 ist also 6.

Das Berechnungsschema, wie es die obige Tabelle zeigt, läßt sich leicht in ein Flußdiagramm umsetzen.

Zunächst wenden wir uns aber noch dem kleinsten gemeinsamen Vielfachen (k.g.V.) zu :

Das k.g.V. von zwei Zahlen Z1 und Z2 kann man dadurch erhalten, daß man das Produkt dieser beiden Zahlen durch den g.g.T. dividiert.

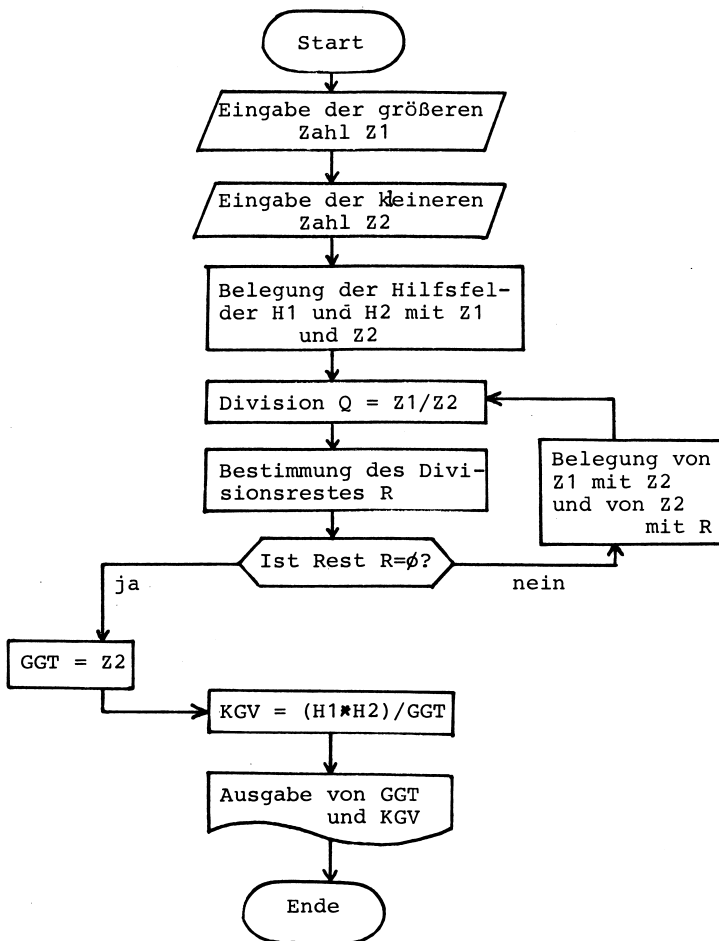
Für die beiden Zahlen Z1 = 78 und Z2 = 42 gilt also :

$$\text{k.g.V.} = \frac{Z1 * Z2}{\text{g.g.T.}} = \frac{78 * 42}{6} = 546$$

Wenn wir in einem BASIC-Programm diese Berechnung durchführen wollen, gemäß der Übersichtstabelle oben aber fortwährend die Feldbesetzungen verändern, müssen vor

Vo8 C64/ Schule	Kapitel 2 : Mathematik Abschnitt 3 : g.g.T. und k.g.V.	Seite 53
<p>der Berechnung des g.g.T. die einzugebenden Werte Z1 und Z2 in zwei Hilfsfeldern H1 und H2 "zwischen gespeichert" werden, um für die Berechnung des k.g.V. dann zur Verfügung zu stehen.</p>		

5. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 3 : g.g.T. und k.g.V.	Seite 55
-----------------------	---	-------------

4. SCHRITT : Programm

```

10 REM M2-GGT UND KGV
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR BESTIMMUNG DES G.G.T. UND"
40 PRINT"DES K.G.V. AUS ZWEI EINGEGEBENEN ZAHLEN"
50 PRINT:PRINT:PRINTTAB(10)"PROF.DR.W.VOSS, 1984":PRINT:PRINT
60 INPUT"GROESSERE ZAHL      : ";Z1
70 PRINT:PRINT
80 INPUT"KLEINERE ZAHL      : ";Z2
90 H1=Z1:H2=Z2
100 Q=Z1/Z2
110 R=Z1-Z2*INT(Q)
120 IF R>0 THEN Z1=Z2:Z2=R:GOTO 100
130 GGT=Z2
140 KGV=(H1*H2)/GGT
150 PRINT:PRINT:PRINT"G.G.T. = ";GGT
160 PRINT:PRINT:PRINT"K.G.V. = ";KGV
170 PRINT:PRINT"ENDE DER BERECHNUNG":END

```

5. SCHRITT : Variablenliste

```

GGT  *  Größter gemeinsamer Teiler
H1   :  Hilfsfeld 1
H2   :  Hilfsfeld 2
KGV  :  Kleinstes gemeinsames Vielfaches
Q    :  Quotient aus beiden Zahlen Z1 und Z2
R    :  Divisionsrest
Z1   :  Größere Zahl
Z2   :  Kleinere Zahl

```

VoB C64/ Schule	Kapitel 2 : Mathematik Abschnitt 3 : g.g.T. und k.g.V.	Seite 56
-----------------------	---	-------------

6. SCHRITT : Programmbeschreibung

- Satz 1 ϕ -5 ϕ : Überschrift und Erläuterungen
- Satz 6 ϕ : Eingabe der einen Zahl
- Satz 7 ϕ : Zwei Leerzeilen
- Satz 8 ϕ : Eingabe der zweiten Zahl
- Satz 9 ϕ : Belegung der beiden Hilfsfelder H1 und H2 mit den eingegebenen Werten, damit diese später noch zur Verfügung stehen (siehe 14 ϕ)
- Satz 10 ϕ : Bestimmung des Quotienten aus beiden eingegebenen Zahlen
- Satz 11 ϕ : Bestimmung des Divisionsrestes durch Benutzung der INT-Funktion
- Satz 12 ϕ : Ist dieser Rest größer als null, wird das Feld Z1 mit der zweiten Zahl Z2 und das Feld Z2 mit dem Rest R belegt und das Programm kehrt zurück zu Satz 10 ϕ
- Satz 13 ϕ : Ist hingegen der Rest gleich null, so steht im Feld Z2 der g.g.T., der in das Feld GGT übertragen wird

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 3 : g.g.T. und k.g.V.	Seite 57
-----------------------	---	-------------

Satz 14 ϕ : Gemäß unserer Vorüberlegungen bestimmt sich dann das k.g.V. aus dem Produkt der beiden Zahlen (die noch in den Feldern H1 und H2 zu finden sind), dividiert durch den g.g.T.

Satz 15 ϕ -16 ϕ : Ausdruck der Ergebnisse mit einigen Leerzeilen dazwischen

Satz 17 ϕ : Beendigung des Programms

⑦. SCHRITT : Programmergebnisse

Nach dem Kommando RUN meldet das Programm auf einem "geräumten" Bildschirm :

PROGRAMM ZUR BESTIMMUNG DES G.G.T. UND
DES K.G.V. AUS ZWEI EINGEGEBENEN ZAHLEN

PROF.DR.W.VOSS, 1984

GROESSERE ZAHL : ?

Geben wir nach Erscheinen dieses Fragezeichens, das ja bekanntlich eine Programmunterbrechung anzeigt, bei-

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 3 : g.g.T. und k.g.V.	Seite 58
<p>spielsweise den Wert 78 ein, so fordert das Programm daraufhin die zweite Zahl an :</p> <p>KLEINERE ZAHL : ?</p> <p>Geben wir zum Beispiel die Zahl 42 ein, so bringt der Rechner die folgenden Ergebnisse auf den Bildschirm :</p> <p>G.G.T. = 6</p> <p>K.G.V. = 546</p> <p>ENDE DER BERECHNUNG</p>		

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 4 : Primzahlenprüfung	Seite 59
-----------------------	---	-------------

2.4 Primzahlenprüfung

① SCHRITT : Vorstellung des Problems

Es soll ein Programm entwickelt werden, welches für eine beliebige eingegebene, positive, ganze Zahl prüft, ob diese eine Primzahl ist oder nicht.

Bekanntlich handelt es sich bei einer Primzahl um eine Zahl, die nur durch 1 oder durch sich selbst ohne Rest teilbar ist.

②. SCHRITT : Problemanalyse

Um zu prüfen, ob eine beliebige Zahl eine Primzahl ist, muß sie durch alle Teiler, die größer als 1, aber kleiner als sie selbst sind, geteilt werden. Sofern bei diesen Divisionen ein Rest auftritt, bzw. wenn das Divisionsergebnis nicht ganzzahlig ist, muß durch einen weiteren Teiler dividiert werden.

Erzielt man bei einer dieser Divisionen ein Ergebnis ohne Rest, so ist die zu prüfende Zahl offenbar keine Primzahl.

Wird bei allen denkbaren Divisionen hingegen ein Rest beobachtet, so ist schließlich festzustellen, daß die zu prüfende Ausgangszahl eine Primzahl ist.

Die Teiler, die im einzelnen durchprobiert werden müssen, beginnen beim Wert 2 und können bei der Hälfte der Ausgangszahl enden; größere Teiler brauchen nicht probiert zu werden, weil dann auf gar keinen Fall mehr ein ganzzahliges Divisionsergebnis erzielt werden kann.

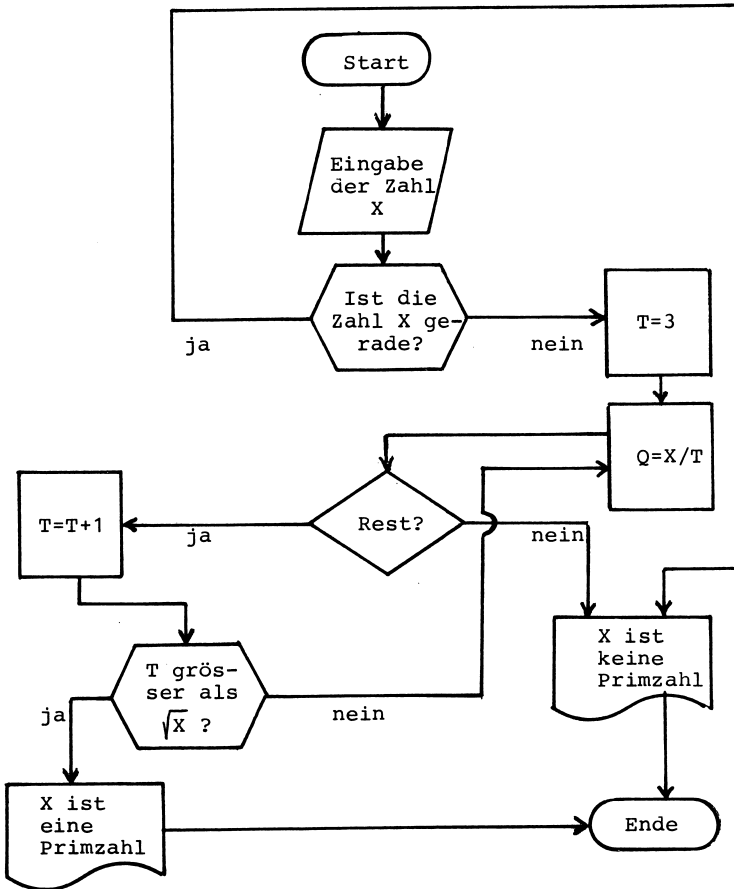
Beispiel : Ausgangszahl $X = 25$

- | | | |
|---------------|-------------|------------------|
| 1. Teiler : 2 | $25/2 = 12$ | Rest 1 |
| 2. Teiler : 3 | $25/3 = 8$ | Rest 1 |
| 3. Teiler : 4 | $25/4 = 6$ | Rest 1 |
| 4. Teiler : 5 | $25/5 = 5$ | Rest \emptyset |
- ===

25 ist keine Primzahl.

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 4 : Primzahlenprüfung	Seite 61
<p>Beispiel : Ausgangszahl $X = 7$</p> <p>1. Teiler : 2 $7/2 = 3$ Rest 1 2. Teiler : 3 $7/3 = 2$ Rest 1</p> <p>Ein weiterer Teiler braucht nicht mehr probiert zu werden, weil das Divisionsergebnis nicht mehr ganzzahlig werden kann.</p> <p>Da bei diesem Beispiel kein Rest \emptyset aufgetreten ist, gilt also :</p> <p>7 ist eine Primzahl.</p> <p>Diese beiden Beispiele zeigen, daß wir Schritt für Schritt den Teiler, mit dem Wert 2 beginnend, um je 1 erhöhen müssen, bis maximal der Wert $X/2$ erreicht ist.</p>		

5. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 4 : Primzahlenprüfung	Seite 63
<p>Dieses Beispiel zeigt übrigens sehr deutlich, daß bei sorgfältiger gedanklicher Durchdringung des Problems das Programm wesentlich vereinfacht und damit auch beschleunigt werden kann :</p> <p>Man kann schon von vornherein feststellen, daß eine <u>gerade</u> Zahl keine Primzahl sein kann, denn sie ist ja auf jeden Fall durch 2 teilbar. Deshalb können gerade Zahlen schon zu Beginn des Prüfprogramms als "Nicht-Primzahlen" ausgeschieden werden.</p> <p>Bei den verbleibenden, den ungeraden Zahlen braucht für eine beliebige Zahl X der Teiler offenbar nicht bis $X/2$ zu laufen, sondern es genügt, wenn wir bis maximal \sqrt{X} die Divisionen probieren. Wenn bis dahin keine Division ohne Rest möglich war, dann wird dies auch bei größeren Teilern nicht mehr möglich sein.</p> <p>Der Leser überlege sorgfältig, warum diese Überlegung richtig ist.</p>		

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 4 : Primzahlenprüfung	Seite 64
-----------------------	---	-------------

④, SCHRITT : Programm

```

10 REM M3-PRIMZAHL
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR PRUEFUNG, OB EINE EINGEGE-"
40 PRINT"BENE ZAHL EINE PRIMZAHL IST ODER NICHT."
50 PRINT:PRINT:PRINTTAB(10)"PROF.DR.W.VOSS, 1984":PRINT:PRINT
60 INPUT"BITTE EINE GANZE ZAHL : ";X
70 T=2
80 IF X/2=INT(X/2) THEN 140
90 Q=X/T
100 IF Q=INT(Q) THEN 140
110 T=T+1
120 IF T<=SQR(X) THEN 90
130 PRINT:PRINT:PRINTX;" IST EINE PRIMZAHL." :GOTO 150
140 PRINT:PRINT:PRINTX;" IST KEINE PRIMZAHL, SONDERN DURCH " :
PRINTT;" TEILBAR."
150 PRINT:PRINT:PRINT"ENDE DER BERECHNUNGEN." :END

```

⑤, SCHRITT : Variablenliste

Q = Quotient aus der Zahl X und dem Teiler T

T = Teiler

X = Einzugebende Zahl, die daraufhin geprüft wird, ob sie Primzahl ist oder nicht

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 4 : Primzahlenprüfung	Seite 65
-----------------------	---	-------------

(6.) SCHRITT : Programmbeschreibung

- Satz 1 ϕ -5 ϕ : Überschrift, Räumung des Bildschirms und Titelausdruck auf dem Bildschirm.
- Satz 6 ϕ : Anforderung der zu prüfenden Zahl.
- Satz 7 ϕ : Belegung des Teilerfeldes T mit 2.
- Satz 8 ϕ : Wenn X durch 2 ohne Rest teilbar sein sollte (dann ist $X/2 = \text{INT}(X/2)$), dann weiter bei Satz 14 ϕ ; wenn nicht, weiter bei Satz 9 ϕ .
- Satz 9 ϕ : Division von X durch T (ergibt Q).
- Satz 10 ϕ : Wenn bei der Division kein Rest aufgetreten ist (dann ist $Q = \text{INT}(Q)$), dann weiter bei Satz 14 ϕ , andernfalls weiter bei Satz 11 ϕ .
- Satz 11 ϕ : Erhöhung des Teilers T um 1.
- Satz 12 ϕ : Solange nach dieser Erhöhung T noch kleiner ist als \sqrt{X} , geht es zurück zum Satz 9 ϕ , andernfalls geht es bei 13 ϕ weiter.
- Satz 13 ϕ : Wird der Satz 13 ϕ erreicht, dann deshalb, weil keine Division ohne Rest vollzogen werden konnte. Dies bedeutet, daß X eine Primzahl sein muß.

Dieses Ergebnis wird ausgegeben und es

VoB C64/ Schule	Kapitel 2 : Mathematik Abschnitt 4 : Primzahlenprüfung	Seite 66
-----------------------	---	-------------

erfolgt ein Sprung zum Satz 15 ϕ .

Satz 14 ϕ : Satz 14 ϕ wird nur in dem Fall erreicht, in dem eine Division ohne Rest gelungen ist.

Dies bedeutet aber dann, daß X keine Primzahl ist, sondern (zumindest) durch den Wert im Feld T teilbar ist.

Dies wird ausgegeben.

Satz 15 ϕ : Beendigung des Programms.

7. SCHRITT : Ergebnisse

Auf die Wiederholung des Titelausdrucks wollen wir nun hier und bei den folgenden Beispielen verzichten.

Nachdem dieser Titelausdruck erfolgt ist, fordert das Programm die zu prüfende Zahl an und speichert sie im Feld X (auch auf die Wiedergabe der entsprechenden Anforderungstexte soll hier und bei den folgenden Beispielen verzichtet werden).

Geben wir z.B. die Zahl 13 ein, so meldet der Rechner :

13 IST EINE PRIMZAHL.

ENDE DER BERECHNUNGEN.

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 4 : Primzahlenprüfung	Seite 67
<p data-bbox="219 352 945 408">Geben wir hingegen die Zahl 27 ein, so meldet der Rechner :</p> <p data-bbox="246 453 736 510">27 IST KEINE PRIMZAHL, SONDERN DURCH 3 TEILBAR.</p> <p data-bbox="219 555 508 576">ENDE DER BERECHNUNGEN.</p>		

2.5

Quadratische Gleichung

①. SCHRITT : Vorstellung des Problems

Das Lösen quadratischer Gleichungen gehört zu den wesentlichen Aufgabenstellungen der Mittelstufenmathematik und bereitet auf komplexere Aufgaben vor.

Die allgemeine Form einer quadratischen Gleichung, die nach der Unbekannten x aufzulösen ist, lautet :

$$(1) \quad ax^2 + bx + c = \emptyset$$

Dividiert man diese Gleichung durch den Koeffizienten a , so erhält man :

$$(2) \quad x^2 + px + q = \emptyset$$

Die Lösung führt zu zwei Werten x_1 und x_2 gemäß der folgenden Lösungsformel :

$$(3) \quad x_{1/2} = -p/2 \pm \sqrt{p^2/4 - q}$$

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 5 : Quadratische Gleichung	Seite 69
-----------------------	--	-------------

② SCHRITT : Problemanalyse

Je nach Konstellation der Koeffizienten a, b und c führt die Lösungsformel (3) zu unterschiedlichen Ergebnissen.

Beispielsweise erhalten wir Resultate außerhalb des reellen Zahlenbereichs, wenn der Wert unter der Wurzel negativ wird. Dieser Wert heißt Diskriminante (D).

Zu Sonderlösungen gelangt man auch z.B. dann, wenn der Koeffizient $a = \emptyset$ ist; dann liegt aber auch eigentlich keine quadratische Gleichung mehr vor.

Deshalb betrachten wir im folgenden nur die "echten" Fälle ($a \neq \emptyset$) und zugleich die im reellen Bereich lösbaren Fälle ($D \geq \emptyset$). Bei allen anderen Fällen soll das Programm eine Abbruchmeldung erzeugen.

Zur Lösung des Problems verwenden wir einfach die Formel (3) in der abgewandelten Form (4) :

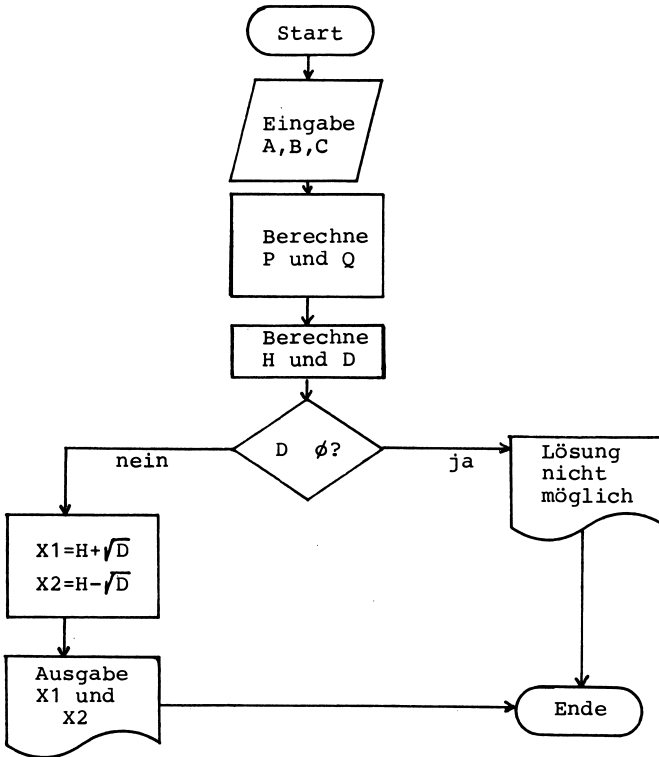
$$(3) \quad x_{1/2} = -p/2 \pm \sqrt{p^2/4 - q}$$

$$(4) \quad x_{1/2} = H \pm \sqrt{D}$$

H steht als
Hilfsgröße
für $-p/2$

Das entsprechende BASIC-Programm muß also zunächst die Größen H und D bestimmen und kann dann, in Abhängigkeit davon, ob $D \geq \emptyset$ oder $D < \emptyset$, die Lösungen bzw. eine Abbruchmeldung ausgeben.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 5 : Quadratische Gleichung	Seite 71
-----------------------	--	-------------

④ SCHRITT : Programm

```

10 REM M4-QUADRATISCHE GLEICHUNG
20 PRINTCHR*(147)
30 PRINT"PROGRAMM ZUR LOESUNG EINER QUADRATISCHEN"
40 PRINTTAB(16)"GLEICHUNG."
50 PRINT:PRINT:PRINTTAB(10)"PROF.DR.W.VOSS, 1984":PRINT:PRINT
60 INPUT"PARAMETER DES QUADR.GLIEDS A : ";A
70 INPUT"PARAMETER DES LIN.GLIEDS B : ";B
80 INPUT"KONSTANTES GLIED C : ";C
90 P=B/A:Q=C/A
100 H=-P/2:D=P*P/4-Q
110 IF D<0 THEN PRINT:PRINT:PRINT"LOESUNG NICHT MOEGLICH.":GOTO 150
120 X1=H+SQR(D):X2=H-SQR(D)
130 PRINT:PRINT:PRINT"ERSTE LOESUNG X1 = ";X1
140 PRINT:PRINT:PRINT"ZWEITE LOESUNG X2= ";X2
150 PRINT:PRINT:PRINT"ENDE DER BERECHNUNGEN":END

```

⑤ SCHRITT : Variablenliste

A = Koeffizient des quadratischen Teils
B = Koeffizient des linearen Teils
C = Konstante in der quadratischen Gleichung

D = Diskriminante
H = $-P/2$
P = B/A
Q = C/A

X1 = erste Lösung
X2 = zweite Lösung

VoB C64/ Schule	Kapitel 2 : Mathematik Abschnitt 5 : Quadratische Gleichung	Seite 72
-----------------------	--	-------------

6. SCHRITT : Programmbeschreibung

- Satz 10-50 : Kommentar und Überschriftausgabe
- Satz 60-80 : Eingabe der Koeffizienten der quadratischen Gleichung (in der Form der Gleichung (1))
- Satz 90 : Bestimmung von P und Q gemäß Gleichung (2)
- Satz 100 : Bestimmung der Hilfsgröße H und der Diskriminate D
- Satz 110 : Wenn D kleiner als null ist, so erfolgt eine Abbruchmeldung auf dem Bildschirm und ein Sprung zu Satz 150
- Satz 120 : Ist die Diskriminante hingegen nicht negativ, so werden die beiden Lösungen X1 und X2 berechnet
- Satz 130-140: Beide Lösungen werden auf dem Bildschirm ausgegeben
- Satz 150 : Beendigung des Programms

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 5 : Quadratische Gleichung	Seite 73
-----------------------	--	-------------

7. SCHRITT : Ergebnisse

Geben wir z.B. ein :

$$A = 4,$$

$$B = 4,$$

$$C = -24,$$

Der Leser sollte
auch andere Werte
erproben !

geht es also um die quadratische Gleichung :

$$4x^2 + 4x - 24 = \emptyset ,$$

so erzeugt das Programm als Ergebnisausgabe :

$$X1 = 2$$

$$X2 = -3$$

ENDE DER BERECHNUNGEN

2.6 Euler'sche Zahl

1. SCHRITT : Vorstellung des Problems

Die Euler'sche Zahl, abgekürzt e , auch Wachstumskonstante genannt, spielt bei mathematischen Reihenentwicklungen, die inhaltlich als Wachstums- oder Fortpflanzungsketten betrachtet werden können, eine wichtige Rolle.

Sie dient darüberhinaus als Basis der sog. natürlichen Logarithmen und wird uns im Kapitel über wirtschaftliche Problemstellungen wieder begegnen.

Wenn man den Wert dieser Zahl e vergessen hat, kann man mit Hilfe eines geeigneten BASIC-Programms diesen Wert ausrechnen, sofern man weiß, wie die Zahl e zustandekommt.

Es handelt sich bei e um den Grenzwert der folgenden Größe, wenn n gegen unendlich strebt :

$$e = \left(1 + \frac{1}{n}\right)^n \quad \text{für } n \text{ gegen unendlich.}$$

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 6 : Euler'sche Zahl	Seite 75
-----------------------	---	-------------

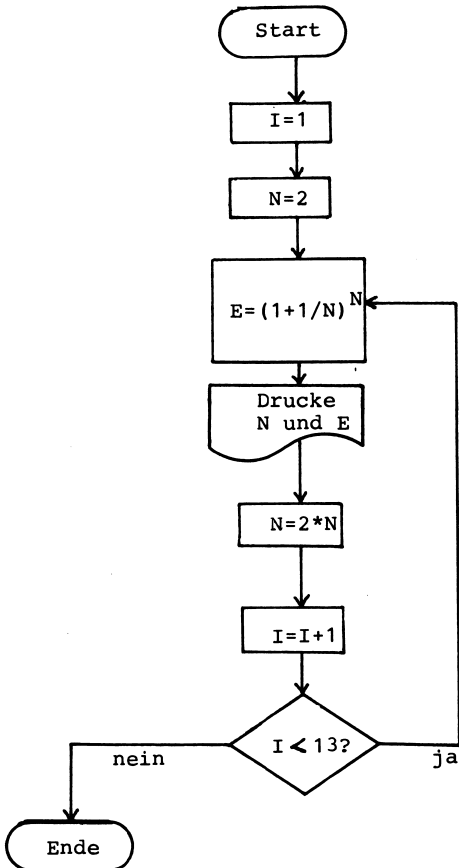
② SCHRITT : Problemanalyse

Die Problemanalyse ist bei dieser Fragestellung sehr einfach, denn es ist lediglich die vorangegangene Formel zu programmieren.

Dieses Programm kann dann für alternative, größer werdende Werte von n durchprobiert werden. Da der Wert e erst für n gegen unendlich erreicht wird, empfiehlt es sich, n rasch wachsen zu lassen, z.B. dadurch, daß man n von Rechendurchgang zu Rechendurchgang verdoppelt.

Dies verlangt allerdings, daß man mit einem n -Wert beginnt, der größer als 1 ist (also z.B. mit $n = 2$), damit bei den Verdoppelungen n nicht beim Wert 1 verharrt.

3. SCHRITT : Flußdiagramm



Es wird davon ausgegangen, daß 12 "Runden" zu durchlaufen sind.

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 6 : Euler'sche Zahl	Seite 77
-----------------------	---	-------------

④ SCHRITT : Programm

```

10 REM M5-EULER
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR BESTIMMUNG DER ZAHL E."
40 PRINT
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984":PRINT
60 N=2
70 PRINT" N", " E":PRINT
80 FOR I=1 TO 12
90 E=(1+1/N)^N
100 PRINTN,E
110 N=2*N
120 NEXT I
130 PRINT"ENDE":END

```

⑤ SCHRITT : Variablenliste

E = Euler'sche Zahl (Näherungswert)

I = Schleifenzählindex

N = Parameter in der Berechnungsformel für die Euler'sche Zahl

Voß C64/ Schule	Kapitel 2: Mathematik Abschnitt 6: Euler'sche Zahl	Seite 78
-----------------------	---	-------------

6. SCHRITT : Programmbeschreibung

Satz 1 ϕ -5 ϕ : Überschrift

Satz 6 ϕ : Belegung des Parameters N mit dem Startwert 2

Satz 7 ϕ : Ausgabe einer Tabellenüberschrift und einer Leerzeile

Satz 8 ϕ -12 ϕ : Schleife über 12 Runden. In jeder Runde wird E berechnet, werden N und E ausgegeben und wird N verdoppelt

Satz 13 ϕ : Ende des Programms

7. SCHRITT : Ergebnisse

Dieses Programm erzeugt eine Tabelle, in der anwachsenden N-Werten die berechneten E-Werte gegenübergestellt werden.

Man sieht bei dieser Gegenüberstellung, daß sich e einem festen Wert annähert.

Wollte man diesem Grenzwert noch näher kommen, dann müßte das Programm noch mehr Schleifendurchläufe erle-

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 6 : Euler'sche Zahl	Seite 79
-----------------------	---	-------------

digen. Man müßte also den Satz 8ø beispielsweise wie folgt ändern :

8ø FOR I = 1 TO 4ø

Es ist allerdings dabei zu beachten, daß bei zu hohen Werten für n die Rechenungenauigkeiten des Computers zu Fehlern führt.

Die sich durch obiges Programm ergebende Tabelle sieht folgendermaßen aus :

N	E
2	2.25
4	2.4414ø625
8	2.56578452
16	2.63792851
32	2.67699ø13
64	2.69734497
128	2.7ø7739ø3
256	2.71299176
512	2.715632ø1
1ø24	2.71695579
2ø48	2.717619ø5
4ø96	2.71795231
ENDE	

2.7 Prozentrechnung

①. SCHRITT : Vorstellung des Problems

Beim Prozentrechnen spielt die Dreisatzrechnung eine entscheidende Rolle. Sie beantwortet zum Beispiel die Frage, welchen Anteil ein bestimmter Betrag B an einer Gesamtgröße G ausmacht. Ein typisches Beispiel kann diese Aufgabenstellung gut verdeutlichen :

Ein Rechnungsbetrag beläuft sich auf netto DM 450.--. Der Kunde muß zusätzlich die sog. Mehrwertsteuer bezahlen, die derzeit 14 % des Nettorechnungsbetrages ausmacht. Somit ergibt sich also :

$$\text{Bruttobetrag} = \text{Nettobetrag} + \text{Steuerbetrag}$$

Die Rechenaufgabe, die sich hier stellt, besteht also darin, den Steuerbetrag und danach den Bruttobetrag zu bestimmen, wenn der Nettobetrag und der Steuer-Prozentsatz gegeben sind.

Natürlich könnte man genauso gut nach dem Nettobetrag fragen, wenn z.B. der Steuersatz und der Bruttobetrag gegeben sind.

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 7 : Prozentrechnung	Seite 81
-----------------------	---	-------------

② SCHRITT : Problemanalyse

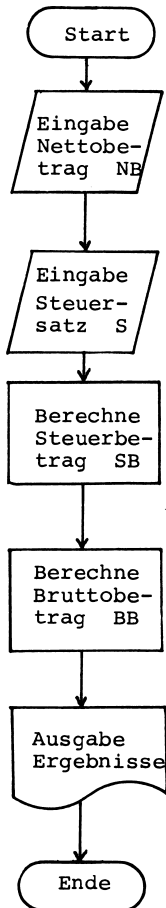
Das oben genannte Beispiel kann mit der Dreisatzrechnung folgendermaßen gelöst werden (und zeichnet uns so den Lösungsweg vor) : ,

Nettobetrag = 100 % entsprechen 450.- DM
Steuerbetrag= 14 % entsprechen x DM

$$x = \frac{450 \cdot 14}{100} = 6.30 \text{ (DM)}$$

Damit wird der Lösungsweg deutlich : Wir müssen in ein allgemein gehaltenes Programm den Nettobetrag eingeben und dazu den Steuersatz (oder allgemein gesprochen, den Prozentsatz, der beliebig sein kann, wenn das Programm generell einsetzbar sein soll).

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 7 : Prozentrechnung	Seite 83
-----------------------	---	-------------

④ SCHRITT : Programm

```

10 REM M6-PROZENT
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR BESTIMMUNG VON PROZENTWER-"
40 PRINTTAB(18)"TEN."
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984":PRINT
60 INPUT"NETTOBETRAG      ":";NB
70 PRINT:INPUT"PROZENTSATZ      ":";SS
80 SB=NB*SS/100
90 BB=NB+SB
95 PRINT:PRINT
100 PRINT:PRINT:PRINT"NETTOBETRAG      ":";NB
110 PRINT:PRINT  "PROZENTBETRAG      ":";SB
120 PRINT:PRINT  "BRUTOBETRAG      ":";BB
130 PRINT:PRINT"ENDE":END

```

⑤ SCHRITT : Variablenliste

BB = Bruttobetrag

NB = Nettobetrag

SB = Prozentbetrag

SS = Prozentsatz

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 7 : Prozentrechnung	Seite 84
-----------------------	---	-------------

(6.) SCHRITT : Programmbeschreibung

Satz 10-50 : Überschrift.

Satz 60-70 : Eingabe von Nettobetrag und Prozentsatz.

Satz 80-90 : Berechnung von Prozentbetrag und Bruttobetrag.

Satz 95 : Zwei Leerzeilen.

Satz 100-120 : Ausgabe der Ergebnisse auf dem Bildschirm.

Satz 130 : Ende des Programms.

(7.) SCHRITT : Ergebnisse

Gibt man beispielsweise als Nettobetrag 200 und als Prozentsatz den Wert 15 ein, so meldet das Programm :

NETTOBETRAG : 200

PROZENTBETRAG : 30

BRUTTOBETRAG : 230

ENDE

Voß C64/ Schule	Kapitel 2 : Mathematik Abschnitt 8 : Runden von Zahlen	Seite 85
-----------------------	---	-------------

28

Das Runden von Zahlen

①. SCHRITT : Vorstellung des Problems

Bei einigen der vorangegangenen Beispiele hat sich schon gezeigt, daß häufig Rechenergebnisse produziert werden mit zu vielen Nachkommastellen.

Wenn beispielsweise mit Geldbeträgen gerechnet wird, wäre es aber sinnvoll, wenn Ergebnisse nur auf zwei Stellen hinter dem Komma ausgegeben würden (genauer gesagt : hinter dem Dezimalpunkt) - wenn möglich, mit korrekter Rundung.

Also sollte z.B. aus 12.342 der gerundete Wert 12.34 werde, aus 12.347 hingegen der Wert 12.35 werden.

Entsprechendes gilt, wenn auf drei Stellen oder auf eine andere Stellenanzahl gerundet werden soll.

2. SCHRITT : Problemanalyse

Bleiben wir bei dem obigen Zahlenbeispiel :

Aus 12.342 soll auf zwei Dezimalstellen gerundet 12.34 werden. Wie ist das zu erreichen ?

Offenbar muß dafür gesorgt werden, daß (in diesem Beispiel) die letzte Stelle "abgeschnitten" wird. Dies kann bekanntlich mit der INT-Funktion gelingen.

Diese Funktion würde aber von der Zahl 12.342 die Stellen .342 abschneiden, so daß nur der Wert 12 übrig bliebe, wenn wir nicht vor ihrem Einsatz den zu runden Wert mit 100 multiplizieren würden.

Aus 12.342 wird so zunächst 1234.2 und durch Einsatz der INT-Funktion dann 1234 .

Wenn wir diese Zahl wieder durch 100 teilen, erhalten wir das gewünschte Ergebnis 12.34 .

Was aber geschieht beim Beispiel 12.347 ? Auch aus dieser Zahl wird mit der eben beschriebenen Prozedur der Wert 12.34, was sicherlich nicht erwünscht ist.

Es muß deshalb dafür gesorgt werden, daß dann, wenn die dritte Dezimalstelle 5 oder größer ist, die zweite Stelle (bisher im Beispiel die Ziffer 4) auf 5 erhöht wird. Dies ist offenbar dadurch zu erreichen, daß vor dem Abschneiden durch die INT-Funktion zu der zu runden Zahl der Wert 0.5 hinzuaddiert wird.

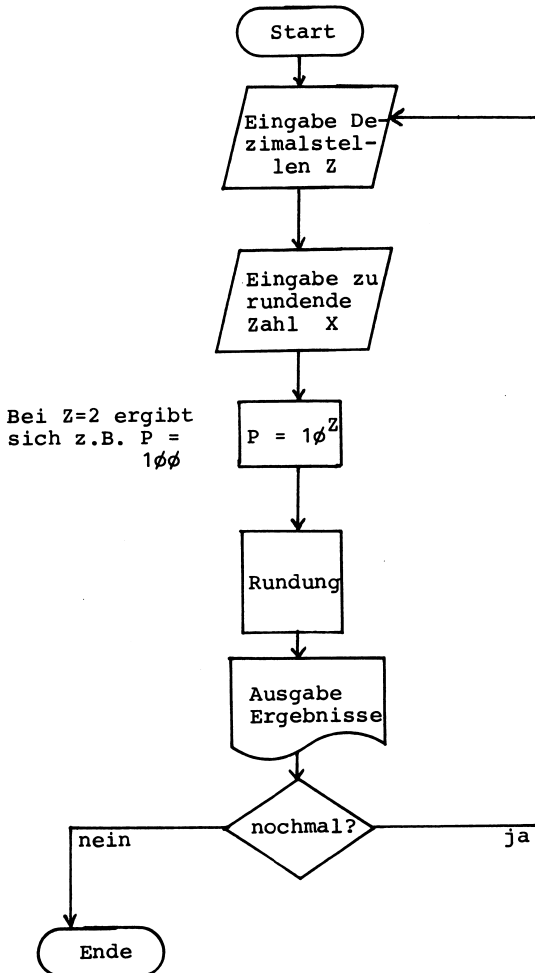
Somit gelangt man zu dem folgenden schematischen Ablauf :

Zahl	*100	+0.5	INT	/100
(1)	(2)	(3)	(4)	(5)
12.342	1234.2	1234.7	1234	12.34
12.347	1234.7	1235.2	1235	12.35

Diese Übersicht zeigt noch einmal schematisch, wie vorzugehen ist, wenn auf zwei Dezimalstellen gerundet werden soll.

Soll hingegen zum Beispiel auf drei Dezimalstellen gerundet werden, so ist in Spalte (2) des obigen Schemas mit 1000 zu multiplizieren und in Spalte (5) durch 1000 zu teilen.

3. SCHRITT : Flußdiagramm



VoB C64/ Schule	Kapitel 2 : Mathematik Abschnitt 8 : Runden von Zahlen	Seite 89
-----------------------	---	-------------

④ SCHRITT : Programm

```

10 REM M7-RUNDEN
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUM RUNDEN VON ZAHLEN AUF BE-"
40 PRINT"LIEBIG VIELE DEZIMALSTELLEN."
50 PRINT:PRINT:PRINTTAB(10)"PROF.DR.W.VOSS, 1984":PRINT
60 PRINT"AUF WIEVIELE STELLEN HINTER DEM DEZIMAL-"
70 INPUT"PUNKT SOLL GERUNDET WERDEN : ";Z
80 PRINT:PRINT:INPUT"ZU RUNDENDE ZAHL : ";X
85 P=10↑Z
90 Y=INT(X*P+.5)/P
100 PRINT:PRINT:PRINT"DIE ZAHL ";X;" LAUTET AUF ";Z;" STELLEN"
110 PRINT"GERUNDET : ";Y
120 PRINT:PRINT:PRINT
130 INPUT"NOCH EINE RECHNUNG ? (J/N) ";A$
140 IF A$="J" THEN PRINTCHR$(147):GOTO 60
150 PRINT:PRINT"ENDE":END

```

⑤ SCHRITT : Variablenliste

A\$ = Antwortstring (Ja, Nein)

P = Korrekturgröße ($P = 10^Z$)

X = Zu rundende Zahl

Y = Gerundete Zahl

Z = Zahl der nach der Rundung gewünschten Dezimalstellen

VoB C64/ Schule	Kapitel 2 : Mathematik Abschnitt 8 : Runden von Zahlen	Seite 90
-----------------------	---	-------------

6. SCHRITT : Programmbeschreibung

- Satz 1 ϕ -5 ϕ : Überschrift
- Satz 6 ϕ -7 ϕ : Angabe der gewünschten Stellenanzahl
- Satz 8 ϕ : Eingabe der zu rundenden Zahl
- Satz 85 : Bestimmung der Korrekturgröße P, wobei gilt : P = 1 $\phi\phi$, wenn auf zwei Stellen gerundet werden soll, P = 1 $\phi\phi\phi$, wenn auf drei Stellen gerundet wird etc.
- Satz 9 ϕ : Rundung gemäß der oben beschriebenen Vorgehensweise
- Satz 1 $\phi\phi$ -12 ϕ : Bildschirmausgabe des Ergebnisses
- Satz 13 ϕ -14 ϕ : Abfrage, ob noch eine weitere Berechnung durchgeführt werden soll
- wenn ja, Säuberung des Bildschirms und zurück zu Satz 6 ϕ (Neueingabe)
- wenn nein, weiter bei Satz 15 ϕ
- Satz 15 ϕ : Ende des Programms

VoB C64/ Schule	Kapitel 2 : Mathematik Abschnitt 8 : Runden von Zahlen	Seite 91
-----------------------	---	-------------

⑦, SCHRITT : Ergebnisse

Geben wir z.B. als Anzahl der gewünschten Dezimalstellen den Wert 1 ein und als zu rundende Zahl die 12.335, so erhalten wir :

DIE ZAHL 12.335 LAUTET AUF 1 STELLEN
 GERUNDET : 12.3

Entsprechend können andere Eingaben erprobt werden.

Natürlich würde es sich auch anbieten, das Programm so zu verschönern, daß es bei dem erwähnten Beispiel nicht ausgibt

... AUF 1 STELLEN...

sondern

... AUF 1 STELLE ...

Der Leser überlege, wie das zu bewerkstelligen sei.

2.9 Probleme der Rechengenauigkeit

Ein Computer wie der Commodore C 64 reserviert für jede reelle Zahl, mit der gerechnet wird, nur eine bestimmte Zahl von Bytes. Dies bedeutet, daß er nur mit beschränkter Genauigkeit arbeiten kann.

Man sieht dies sehr deutlich, wenn man in dem folgenden Programm "große" Werte vorgibt, und diese sich wieder ausdrucken läßt :

```
1Ø REM ZAHLENGENAUIGKEIT
2Ø A=12345678
3Ø B=123456789
4Ø C=1234567891
5Ø D=12345678912
6Ø ?A:?B:?C:?D
7Ø END
```

Als Ergebnisse erhalten wir :

```
12345678
123456789
1.23456789E+Ø9
1.23456789E+1Ø
```

Bekommt der Rechner also Zahlen mit mehr als 9 Ziffern (Beispiel C und D), dann wandelt er sie bei der Ausgabe in die sog. Exponentialdarstellung um und druckt z.B. für C :

VoB C64/ Schule	Kapitel 2 : Mathematik Abschnitt 9 : Rechengenauigkeit	Seite 94
-----------------------	---	-------------

nur "abgeschnitten", sondern korrekt gerundet.

Das Hauptproblem dabei ist nun folgendes :

Wenn wir beispielsweise zum Wert D in obigem Programm den Wert 1 addieren, etwa in der neuen Programmzeile 55:

```
55 E = D + 1
```

und auch diesen Wert E ausdrucken lassen :

```
65 ?E
```

so erkennen wir keinen Unterschied in den Werten für D und E - der Rechner rechnet falsch !

Auf dieses Problem muß man also immer dann achten, wenn man z.B. sehr große mit sehr kleinen Zahlen (und umgekehrt) verrechnen will.

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 1 : BASIC	Seite 95
-----------------------	---	-------------



Kapitel 3 : Chemie
=====

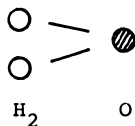
3.1 Zusätzliche BASIC-Anweisungen

Will man Programme aus dem Bereich der Chemie programmieren, so kann man hervorragend die Graphik- und Farbmöglichkeiten des Commodore C 64 ausnutzen - etwa zur graphischen Darstellung von Molekülen oder von chemischen Verbindungen.

Beispielsweise kann man das Wassermolekül



das ja bekanntlich aus zwei Wasserstoffatomen (H) und einem Sauerstoffatom (O) zusammengesetzt ist, anschaulich folgendermaßen darstellen :



Um solche Darstellungen erzeugen zu können, benötigt man zusätzliche Anweisungen. Diese beziehen sich zum einen auf die Farberzeugung und den Farbenwechsel innerhalb eines laufenden Programms und zum anderen auf die Verwendung der auf der Tastatur des C 64 zur Verfügung stehenden Graphiksymbole (auf die Möglichkeiten der sog. hochauflösenden Graphik kommen wir erst in einem späteren Kapitel zu sprechen).

Benutzung der Graphikzeichen der Tastatur

Zum "Zeichnen" von Molekülstrukturen eignet sich besonders gut z.B. das Symbol \bigcirc , welches auf der Vorderseite der Q-Taste zu finden ist.

Mit folgenden Statements kann dieser Kreis gezeichnet werden :

<pre>1Ø PRINT " SHIFT und Q "</pre> <pre>2Ø END</pre>

Der Satz 1Ø erscheint auf dem Bildschirm folgendermaßen :

VoB C64/ Schule	Kapitel 3 : Chemie Abschnitt 1 : BASIC	Seite 97
-----------------------	---	-------------

1Ø PRINT "○"

Der Leser sollte dieses kleine Programm erproben.

Erzeugung von Farben im Programm

Die Zifferntasten in der oberen Reihe der Tastatur sind zusätzlich mit Farben belegt. Daher kann beispielsweise mit der Taste 3 unter gleichzeitiger Betätigung der CTRL -Taste die Farbe Rot erzeugt werden.

Der Leser erprobe das folgende kleine Programm :

1Ø PRINT " CTRL und 3 DIES IST ROT"
2Ø END

Der Satz 1Ø erscheint auf dem Bildschirm als :

1Ø PRINT " £ DIES IST ROT"

Unterschiedliche Farben erzeugen wir mit dem Commodore C 64 dadurch, daß wir

- entweder die CTRL -Taste zusammen mit

einer Zifferntaste,

- oder die **C**-Taste zusammen mit einer Zifferntaste benützen.

Die folgende Tabelle zeigt, welche Farbmöglichkeiten zur Verfügung stehen :

CTRL	+ 1	schwarz	C<	+ 1	orange
	+ 2	weiß		+ 2	braun
	+ 3	rot		+ 3	hellrot
	+ 4	türkis		+ 4	grau 1
	+ 5	violett		+ 5	grau 2
	+ 6	grün		+ 6	hellgrün
	+ 7	blau		+ 7	hellblau
	+ 8	gelb		+ 8	grau 3

Will man aus einer speziellen Farbe wieder "aussteigen", und zurück in den Normalmodus (hellblau auf blau), so ist dies am einfachsten dadurch möglich, daß man die Tasten **RUN/STOP** und **RESTORE** gemeinsam drückt.

Weiterhin benötigen wir in diesem Kapitel und in den meisten der folgenden auch einige Anweisungen, die es uns erlauben, größere Informationsmengen geschickter als dies per LET- oder INPUT-Statement möglich wäre, in den Rechnen zu bekommen.

Dazu eignet sich besonders gut das DATA-Statement in Verbindung mit dem READ-Statement.

VoB C64/ Schule	Kapitel 3 : Chemie Abschnitt 1 : BASIC	Seite 99
-----------------------	---	-------------

Statement 11:

```
nn DATA Wert1,Wert2,Wert3,...
```

Statement 12:

```
nn READ Variable1,Variable2,Variable3,...
```

Beispiel :

```
1Ø REM DATENEINGABE
2Ø DATA 46ØØ, DORTMUND, 463Ø, BOCHUM, 8ØØØ
3Ø DATA MUENCHEN, 2ØØØ, HAMBURG
4Ø READ P1, O1Ø, P2, O2Ø, P3, O3Ø, P4, O4Ø
5Ø ?P1, O1Ø: ?P2, O2Ø: ?P3, O3Ø: ?P4, O4Ø
6Ø END
```

Der Leser erprobe dieses Programm.

Bei großen Datenbeständen empfiehlt es sich, nur einen Variablennamen zu verwenden, der dann aber mehrere Werte erfaßt. Dies gelingt durch Verwendung indizierter Variablen, wie man sie ja zum Beispiel aus der Mathematik kennt. Üblicherweise schreibt man in der Mathematik :

x_1, x_2, x_3, \dots allgemein x_i

Dies geht auch in entsprechender Weise in der Programmiersprache BASIC :

$X(1), X(2), X(3), \dots$ allgemein $X(I)$

(generell :

Variablenname(Indexname))

Bei Verwendung solcher indizierten Variablen muß aber dem Rechner zu Beginn des entsprechenden Programms mitgeteilt werden, wieviele Speicherplätze für die jeweilige Variable freigehalten werden müssen.

Dies leistet das DIM-Statement :

Statement 13:

nn DIM Var.name1(Zahl1), Var.name2(Zahl2), ...

Beispielsweise hält das Statement

10 DIM X(20)

für die Variable X 21 Speicherplätze frei.

Voß C64/ Schule	Kapitel 3: Chemie Abschnitt 1: BASIC	Seite 101
-----------------------	---	--------------

Es handelt sich dabei um die Plätze $X(\emptyset)$, $X(1)$, $X(2)$, $X(3)$, ... , $X(2\emptyset)$.

Ergänzend ist darauf hinzuweisen, daß zum Beispiel auch doppelte Indizierungen möglich sind :

Das Statement

```
1 ̸ DIM Z(2,3)
```

hält für die Variable Z $3 \times 4 = 12$ Plätze frei, die man sich in drei Zeilen und vier Spalten aufgeteilt vorstellen kann :

$Z(\emptyset, \emptyset)$	$Z(\emptyset, 1)$	$Z(\emptyset, 2)$	$Z(\emptyset, 3)$
$Z(1, \emptyset)$	$Z(1, 1)$	$Z(1, 2)$	$Z(1, 3)$
$Z(2, \emptyset)$	$Z(2, 1)$	$Z(2, 2)$	$Z(2, 3)$

Die allgemeine Schreibweise lautet in diesem Fall :

$$Z(I, J)$$

Der erste Index (hier I) ist der sog. Zeilenindex, der zweite (hier J) der Spaltenindex.

Zum Abschluß muß erwähnt werden, daß mit den READ-Statements eines Programms immer in Abfolge auf die vorhandenen DATA-Statements zugegriffen wird. Dies bedeutet, daß ein zweites READ-Statement dort in den vorhandenen DATA-Statements fortfährt zu lesen, wo

das vorangegangene READ-Statement aufgehört hat.

Häufig ist es nun allerdings sinnvoll, daß mit einem zweiten (oder sonst folgenden) READ-Statement die DATA-Liste wieder von Anfang an gelesen wird.

Um dies zu ermöglichen, benötigt man ein weiteres Statement :

Statement 14:

nn RESTORE

Dieses Statement bewirkt, daß der "Lesezeiger" wieder an den Anfang des ersten DATA-Statements gestellt wird, so daß die Daten mit dem folgenden READ-Statement wieder von Anfang an gelesen werden.

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 2 : Wassermolekül	Seite 103
-----------------------	---	--------------

3.2 Das Wassermolekül

①. SCHRITT : Vorstellung des Problems

Es soll mit Hilfe eines BASIC-Programms ein graphisches Abbild des Wassermoleküls (H_2O) erzeugt werden (siehe auch Abschnitt 3.1). Die Wasserstoffatome sollen z.B. als grüne Kügelchen, das Sauerstoffatom als rotes Kügelchen präsentiert werden.

Mit einem derartigen Programm hätte man dann einen Grundbaustein auch für komplexere Programme graphischer Darstellungen dieser Art zur Verfügung.

②. SCHRITT : Problemanalyse

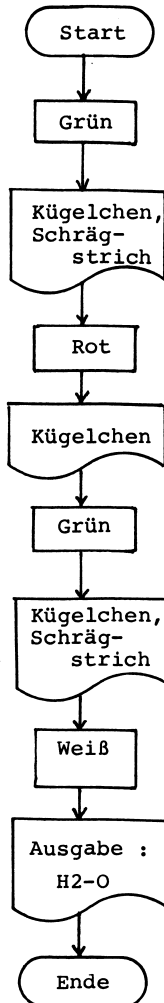
Wenn man das Bild, welches durch das Programm entstehen soll, schon vor Augen hat, dann ist in diesem Beispiel die Problemanalyse sehr einfach :

Es soll genau das Bild erzeugt werden, das wir schon zu Beginn des Abschnitts 3.1 gezeichnet haben.

Wir erkennen, daß dieses Bild sich aus vier Zeilen aufbaut :

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 2 : Wassermolekül	Seite 104
<p>1. Zeile : Ein grünes Kugelchen und ein Schrägstrich von links oben nach rechts unten.</p> <p>2. Zeile : Ein rotes Kugelchen (nach rechts versetzt).</p> <p>3. Zeile : Ein grünes Kugelchen unter dem der ersten Zeile und ein Schrägstrich von links unten nach rechts oben.</p> <p>4. Zeile : Text : H₂O .</p>		

3. SCHRITT : Flußdiagramm



VoB C64/ Schule	Kapitel 3 : Chemie Abschnitt 2 : Wassermolekül	Seite 106
-----------------------	---	--------------

④ SCHRITT : Programm

```

10 REM C1-H2O
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR GRAPHISCHEN DARSTELLUNG "
40 PRINT"      DES WASSERMOLEKUELS H2-O"
50 PRINT:PRINT:PRINTTAB(10)"PROF.DR.W.VOSS, 1984":PRINT
60 PRINT
70 PRINTTAB(10)"  0";
80 PRINTTAB(12)"  \"
90 PRINTTAB(14)"  0"
100 PRINTTAB(10)"  0";
110 PRINTTAB(12)"  \/"
120 PRINT"  "
130 PRINT:PRINT:PRINTTAB(10)"H2- O"
140 PRINT:PRINT"ENDE":END

```

⑤ SCHRITT : Variablenliste

In diesem Programm werden keine Variablen benutzt. Deshalb entfällt hier dieser Arbeitsschritt.

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 2 : Wassermolekül	Seite 107
-----------------------	---	--------------

6. SCHRITT : Programmbeschreibung

- Satz 1 ϕ -6 ϕ : Überschrift
- Satz 7 ϕ : Umschalten auf Grün und Ausgabe eines K gelchens in Spalte 1 ϕ ; Unterdr ckung des Zeilenvorschubs.
- Satz 8 ϕ : Ausgabe eines Schr gstrichs in Spalte 12
- Satz 9 ϕ : Umschalten auf Rot und Ausgabe eines K gelchens in Spalte 14
- Satz 1 $\phi\phi$: Umschalten auf Gr n und Ausgabe eines K gelchens in Spalte 1 ϕ ; Unterdr ckung des Zeilenvorschubs
- Satz 11 ϕ : Ausgabe eines Schr gstrichs in Spalte 12
- Satz 12 ϕ : Umschalten auf Wei .
- Satz 13 ϕ : Ausgabe von H2- O ab Spalte 1 ϕ nach zwei Leerzeilen
- Satz 14 ϕ : Ende des Programms

⑦, SCHRITT : Ergebnisse

Die Bildschirmausgabe dieses Programms entspricht dem Bild, das wir in Abschnitt 3.1 schon vorgestellt haben. Eine weitere Präsentation ist deshalb hier entbehrlich.

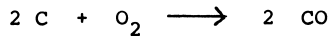
3.3 Reaktionsgleichung

①. SCHRITT : Vorstellung des Problems

Auch chemische Reaktionen lassen sich anschaulich graphisch darstellen, wie am Beispiel der für die Stahlerzeugung notwendigen Prozesse gezeigt werden kann.

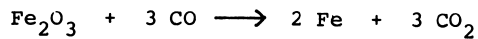
Die Stahlerzeugung in den industrialisierten Ländern beginnt damit, daß im Hochofen zunächst Roheisen produziert wird. Diese Produktion geht in folgenden Schritten vor sich :

1. Schritt : Erhitzte Luft wird mit Koks zu Kohlenmonoxid verbrannt :



Koks	Sauerstoff	Kohlenmonoxid
------	------------	---------------

2. Schritt : Das heiße Kohlenmonoxid (CO) steigt im Hochofen auf und reagiert dort mit Eisenoxid (Fe_2O_3) :



Eisen- oxid	Kohlen- monoxid	Eisen	Kohlen- dioxid
----------------	--------------------	-------	-------------------

Damit steht dann Roheisen für die weitere Verwendung zur Verfügung.

Es soll nun ein BASIC-Programm vorgestellt werden, welches diese chemischen Reaktionsbeziehungen veranschaulicht.

Bei der Darstellung der Atome und der Moleküle bedienen wir uns der gleichen graphischen Möglichkeiten wie im Beispiel zuvor.

② SCHRITT : Problemanalyse

Mit der obigen Beschreibung und unter Rückgriff auf die Überlegungen im vorangegangenen Beispiel ist im Grunde die Problemanalyse schon geleistet, so daß weitere Ausführungen an dieser Stelle entbehrlich sind.

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 3 : Reaktionsgleichung	Seite 111
-----------------------	--	--------------

3. SCHRITT : Flußdiagramm

Auch das Flußdiagramm ist hier nicht mehr erforderlich, wobei die gleichen Gründe maßgeblich sind wie im Schritt zuvor.

4. SCHRITT : Programm

```

10 REM C2-CHEM.REAKTION
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR DARSTELLUNG EINER CHEMI-
40 PRINT"          SCHEN REAKTION"
50 PRINT:PRINT:PRINTTAB(10)"PROF.DR.W.VOSS, 1984":PRINT
60 PRINT"☐";:PRINT"☐"
70 PRINT:PRINTTAB(5)+"";
80 PRINTTAB(10)"☐";:PRINT"☐":PRINTTAB(13)"2"
90 PRINT"☐";:PRINT"☐"
100 FOR I=1 TO 1000:NEXTI
105 PRINT"☐"
110 PRINTTAB(5)"|":FOR I=1 TO 500:NEXTI:PRINTTAB(5)"|"
112 FORI=1TO500:NEXT I
115 PRINTTAB(4)"\";"|";"/"
117 FORI=1TO500:NEXT I
120 PRINT:PRINTTAB(5)"☐";"☐";"☐CO"
125 PRINT"☐"
130 PRINT:PRINTTAB(5)"☐";"☐";"☐CO"
135 PRINT"☐"
140 PRINT:PRINT"2 C + O = 2 CO"
142 PRINTTAB(9)"2"
150 PRINT"BITTE CONT EINGEBEN":STOP
160 PRINTCHR$(147)
170 PRINTTAB(5)"☐";:PRINTTAB(20)"☐";"☐";"☐CO"
180 PRINTTAB(4)"☐"
190 PRINTTAB(5)"☐";:PRINTTAB(8)"☐FE";:PRINT"☐";:PRINT"☐";:
PRINT"☐";

```

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 3 : Reaktionsgleichung	Seite 112
<pre> 195 PRINTTAB(15)"⊕+"; 200 PRINTTAB(20)"⊕⊕"; "⊕⊕"; "⊕CO" 210 PRINTTAB(4)"⊕⊕";:PRINTTAB(10)"⊕⊕";:PRINTTAB(12)"⊕⊕" 220 PRINTTAB(5)"⊕⊕";:PRINTTAB(20)"⊕⊕"; "⊕⊕"; "⊕CO" 225 FOR J=1TO3 230 FORI=1TO500:NEXTI:PRINTTAB(10)"⊕ " 240 NEXT J 250 PRINTTAB(9)"\";" ";"/";:PRINTTAB(14)"FE 0 + 3CO = 2FE + 3CO" 252 PRINTTAB(16)"2";:PRINTTAB(18)"3";:PRINTTAB(36)"2" 270 PRINTTAB(21)"⊕⊕" 280 PRINTTAB(20)"⊕⊕";:PRINTTAB(22)"⊕CO" 290 PRINTTAB(21)"⊕⊕";:PRINTTAB(24)"⊕⊕" 300 PRINTTAB(4)"⊕⊕";:PRINTTAB(7)"⊕FE" 310 PRINTTAB(21)"⊕⊕" 320 PRINTTAB(15)"⊕+";:PRINTTAB(20)"⊕⊕";:PRINTTAB(22)"⊕CO" 330 PRINTTAB(21)"⊕⊕";:PRINTTAB(24)"⊕⊕" 340 PRINTTAB(4)"⊕⊕";:PRINTTAB(7)"⊕FE" 350 PRINTTAB(21)"⊕⊕" 360 PRINTTAB(20)"⊕⊕";:PRINTTAB(22)"⊕CO" 370 PRINTTAB(21)"⊕⊕";:PRINTTAB(24)"⊕⊕" 380 PRINT"⊕ENDE":END </pre>		

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 3 : Reaktionsgleichung	Seite 113
-----------------------	--	--------------

5. SCHRITT : Variablenliste

I = Schleifenindex

J = Schleifenindex

6. SCHRITT : Programmbeschreibung

Satz 1 ϕ -5 ϕ : Überschrift

Satz 6 ϕ : Ausgabe braunes K ugelchen, wei es C

Satz 7 ϕ : Ausgabe + - Zeichen in Spalte 5 nach
einer Leerzeile

Satz 8 ϕ : Ausgabe von zwei roten K ugelchen, einem
wei en O und der Ziffer 2

Satz 9 ϕ : wie Satz 6 ϕ

Satz 1 $\phi\phi$: Warteschleife

Satz 1 ϕ 5 : Umschalten auf Wei 

Satz 11 ϕ : Zeichnen eines senkrechten Strichs mit
zeitlicher Verz ogerung  ber zwei Zeilen

Satz 112 : wie Satz 1 $\phi\phi$

Satz 115 : Zeichen einer Reaktionspfeil-"Spitze"

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 3 : Reaktionsgleichung	Seite 114
	<p>Satz 117 : Warteschleife</p> <p>Satz 12ø : Zeichnen eines braunen K�gelchens, eines roten K�gelchens und Ausgabe von CO in Wei�</p> <p>Satz 125 : Umschalten auf Hellblau</p> <p>Satz 130 : wie Satz 12ø</p> <p>Satz 135 : Umschalten auf Wei�</p> <p>Satz 14ø-142 : Ausgabe von $2\text{C} + \text{O}_2 = 2\text{CO}$</p> <p>Satz 15ø : Ausgabe von "Bitte cont eingeben" (in Hellblau) und Stoppen des Programms</p> <p>Satz 16ø : R�umen des Bildschirms zur Vorbereitung der graphischen Darstellung der zweiten Reaktionsgleichung</p> <p>Satz 17ø-37ø : Im Prinzip nach dem gleichen Muster wie oben ausf�hrlich beschrieben, wird in diesem Programmteil die zweite Reaktionsgleichung graphisch umgesetzt.</p> <p>Eine detaillierte Erl�uterung d�rfte sich angesichts der obigen Ausf�hrungen er�brigen</p> <p>Satz 38ø : Ende des Programms</p>	

VoB C64/ Schule	Kapitel 3 : Chemie Abschnitt 3 : Reaktionsgleichung	Seite 115
-----------------------	--	--------------

7. SCHRITT : Ergebnisse

Das vorgestellte Programm erzeugt nacheinander zwei graphische Bilder auf dem Bildschirm; das zweite Bild erscheint, wenn der Programmbenutzer das Kommando CONT eingegeben hat.

Es dürfte entbehrlich sein, diese beiden Bilder hier vorzustellen, insbesondere auch deshalb, weil hier keine farbliche Darstellung möglich ist. Sinnvoller ist es deshalb, dieses Programm direkt an einem Farbfernseher zu erproben.

3.4 Stöchiometrisches Rechnen

① SCHRITT : Vorstellung des Problems

Beim stöchiometrischen Rechnen geht es darum, die Stoffmengen zu quantifizieren, die bei chemischen Reaktionen umgesetzt werden. Es könnte damit also zum Beispiel die folgende Frage beantwortet werden :

Wieviel Gramm Schwefel sind notwendig, um 100 Gramm Zink vollständig zu Zinksulfid umzusetzen ? (Schwefel = S; Zink = Zn; Zinksulfid = ZnS)

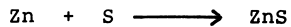
Atomgewicht von Schwefel : 32.1

Atomgewicht von Zink : 65.4

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 4 : Stöchiometrie	Seite 117
-----------------------	---	--------------

(2.) SCHRITT : Problemanalyse

Zur Lösung des gestellten Problems ist von der zuständigen Reaktionsbeziehung auszugehen :



Wenn wir die sog. Molmassen des gesuchten und des gegebenen Stoffs aufeinander beziehen, erhalten wir :

32.1 g Schwefel sind notwendig, um 65.4 g
Zink umzusetzen (s.o.).

Nun gilt offenbar :

$$x/100 = 32.1/65.4$$

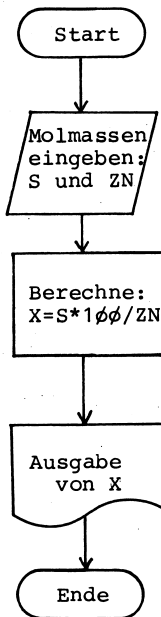
oder

$$x = 3210/65.4$$

Dabei ist x der gesuchte Wert.

3. SCHRITT : Flußdiagramm

Das Flußdiagramm ist bei dieser Fragestellung wieder sehr einfach :



Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 4 : Stöchiometrie	Seite 119
-----------------------	---	--------------

④ 4. SCHRITT ; Programm

```

10 REM C3-CHEM.RECHNEN
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUM STOECHIOMETRISCHEN RECHNEN"
40 PRINT:PRINT
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984":PRINT
60 PRINT:PRINT:PRINT"BEISPIEL :":PRINT:PRINT
70 PRINT"ZINK + SCHWEFEL = ZINKSULFID : "
80 PRINT:PRINT"  ZN +    S    =    ZN S":PRINT:PRINT
90 PRINT:PRINT"WIEVIEL GRAMM SCHWEFEL SIND NOTWENDIG,"
100 PRINT"UM 100 G ZINK VOLLSTÄNDIG ZU ZINKSUL-"
110 PRINT"FID UMZUSETZEN ? ":PRINT:PRINT
120 INPUT"1. ELEMENT : ";A$
130 INPUT"2. ELEMENT : ";B$
140 PRINT:PRINT:PRINT
150 PRINT"ATOMGEWICHT VON ";A$;PRINTTAB(30)":":INPUT A
160 PRINT"ATOMGEWICHT VON ";B$;PRINTTAB(30)":":INPUT B
170 X=B*100/A
180 PRINT:PRINT:PRINT"BENÖTIGTE MENGE VON ";B$; " : " :
      PRINT:PRINTTAB(5)X;" G"
190 PRINT:PRINT:PRINT"ENDE DER BERECHNUNG":END

```

⑤ 5. SCHRITT ; Variablenliste

A = Atomgewicht Stoff A
A\$ = Bezeichnung Stoff A

B = Atomgewicht Stoff B
B\$ = Bezeichnung Stoff B

X = Gesuchte Menge

6. SCHRITT : Programmbeschreibung

Satz 10-11 : Überschrift und Beispielsangabe mit
der in diesem Programm zu behandelnden
Fragestellung.

Satz 12-13 : Eingabe der Namen der beiden Stoffe

Satz 14 : 3 Leerzeilen

Satz 15-16 : Ausgabe der Namen der Stoffe und An-
forderung der Atomgewichte dieser
Stoffe

Satz 17 : Berechnung der gesuchten Menge

Satz 18 : Ausgabe des Ergebnisses (zuzüglich eini-
ger Leerzeilen, auf die aber zukünftig
nicht immer gesondert aufmerksam ge-
macht werden soll).

Satz 19 : Ende des Programms

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 4 : Stöchiometrie	Seite 121
-----------------------	---	--------------

(7) SCHRITT : Ergebnisse

Nach den erläuternden Erklärungen fordert das Programm den Namen des ersten Elements an :

Geben wir z.B. ein "Schwefel", so fordert das Programm den Namen des zweiten Elements, das in die Rechnung eingehen soll, an :

Geben wir z.B. ein "Zink", so fordert das Programm die Atomgewichte für Schwefel und Zink an.

Wenn wir daraufhin eingeben die Werte 32.1 und 65.4, dann liefert das Programm das folgende Ergebnis :

BENOETIGTE MENGE VON ZINK :

49.ø825688 G.

ENDE DER BERECHNUNG

Es versteht sich, daß bei anderen chemischen Prozessen (z.B. auch dann, wenn mehr als zwei Stoffe an den in Frage stehenden Reaktionen beteiligt sind), das obige Programm unter Umständen sehr grundlegend verändert werden muß.

Hier kam es nur darauf an, beispielhaft ein derartiges Programm einmal vorzustellen.

3.5

Das Periodensystem der Elemente

1. SCHRITT : Vorstellung des Problems

Ein Rechner kann nicht nur dafür eingesetzt werden, Rechnungen durchzuführen, wie das zum Beispiel im vorangegangenen Abschnitt der Fall war, oder anschauliche graphische Abbilder zu erzeugen (wie im Abschnitt davor), sondern er kann - gerade im schulischen Bereich - als "Auskunftei" benutzt werden.

Am Beispiel des Periodensystems der Elemente kann verdeutlicht werden, wie ein solches "Nachschlageprogramm", das gewissermaßen auch als Trainingsprogramm verstanden werden kann, aussehen könnte.

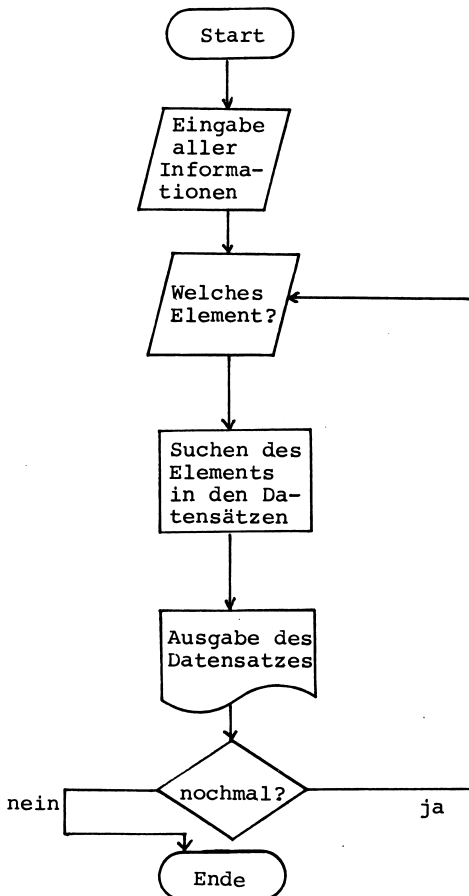
Beispielsweise könnte der Rechenbenutzer daran interessiert sein, nach Eingabe eines Elementnamens (z.B. Schwefel) vom Rechner wesentliche Informationen über dieses chemische Element zu erfahren.

Solche Informationen sind etwa :

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 5 : Periodensystem	Seite 123
<p style="text-align: center;"> 1. Chemisches Zeichen, 2. Ordnungszahl im Periodensystem 3. Atomgewicht, 4. Wichte, 5. Schmelzpunkt, 6. Siedepunkt, 7. Gruppenzugehörigkeit (I-VIII, ϕ), 8. Nebengruppenzugehörigkeit (a,b) 9. Elektronenschalen (K-Q) </p> <p style="text-align: center;"> 2. SCHRITT : Problemanalyse </p> <p> Die Entwicklung eines entsprechenden Computerprogramms, das die interessierenden Auskünfte bereitstellen kann, bereitet keine besonderen Schwierigkeiten. </p> <p> Über DATA und READ können die Gesamtinformationen eingegeben werden. Der Benutzer gibt dann über INPUT an, über welches Element er Informationen wünscht und das Programm stellt dann den gesamten Datensatz für dieses Element zur Verfügung. </p> <p> Schließlich kann abgefragt werden, ob für ein weiteres chemisches Element diese Informationen gewünscht werden. </p>		

3. SCHRITT : Flußdiagramm

Entsprechend einfach wie die Problemanalyse im vorangegangenen Abschnitt ist auch das Flußdiagramm :



VoB C64/ Schule	Kapitel 3: Chemie Abschnitt 5: Periodensystem	Seite 125
-----------------------	--	--------------

④ SCHRITT : Programm

```

10 REM C4-ELEMENTE
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR ANGABE WESENTLICHER EIGEN-"
40 PRINTTAB(5)"SCHAFTEN CHEMISCHER ELEMENTE."
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984":PRINT
55 PRINT:PRINT:PRINT:PRINT"MOMENT BITTE":PRINT:PRINT
60 N=52:REM ANZAHL DER AUFGENOMMENEN ELEMENTE
70 DIM N$(N,10)
75 FOR J=1 TO 7
80 FOR I=1 TO N
85 READ N$(I,J)
90 NEXT I
95 NEXTJ
100 FOR I=1TON
105 READ N$(I,8),N$(I,9),N$(I,10)
108 NEXT I
110 PRINT:INPUT"WELCHES ELEMENT (NAME BITTE) ";B$
120 FOR I=1TON:IF B$=N$(I,1) THEN 160
130 NEXT I
140 PRINT:PRINT"DIESES ELEMENT IST IM DATENBESTAND"
150 PRINT"NICHT VORHANDEN.":GOTO 290
160 PRINTCHR$(147)
170 PRINT"1";N$(I,1):PRINTTAB(23)N$(I,2)
180 PRINT"2":PRINT
190 PRINT"ORDNUNGSZAHL           : ";N$(I,3)
200 PRINT"GRUPPE                 : ";N$(I,8)
210 PRINT"NEBENGRUPPE           : ";N$(I,9)
220 PRINT"SCHALE                 : ";N$(I,10)
230 PRINT:PRINT
240 PRINT"ATOMGEWICHT           : ";N$(I,4)
250 PRINT"WICHTE                 : ";N$(I,5)
260 PRINT
270 PRINT"SCHMELZPUNKT          : ";N$(I,6)
280 PRINT"SIEDEPUNKT            : ";N$(I,7)
290 PRINT:PRINT:INPUT"EIN ANDERES ELEMENT (J/N) ";A$
300 IF A$="J" THEN PRINTCHR$(147):GOTO 110
310 PRINT:PRINT"ENDE IN 310":END

```

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 5 : Periodensystem	Seite 126
<p>495 REM DATEN BEGRIFFE</p> <p>500 DATA ALUMINIUM,ANTIMON,ARGON,ARSEN,BARIUM,BLEI,BOR,BROM,CHLOR,CHROM</p> <p>510 DATA EISEN,FLUOR,GOLD,HELIUM,IRIDIUM,JOD,KADMIUM,KALIUM,KALZIUM,KOBALT</p> <p>520 DATA KOHLENSTOFF,KUPFER,LITHIUM,MAGNESIUM,MANGAN,MOLYBDAEN,NATRIUM</p> <p>530 DATA NEON,NICKEL,PHOSPHOR,PLATIN,PLUTONIUM,QUECKSILBER,RADIUM,RADON</p> <p>540 DATA SAUERSTOFF,SCHWEFEL,SELEN,SILBER,SILIZIUM,STICKSTOFF</p> <p>550 DATA STRONTIUM,THORIUM,TITAN,URAN,VANADIUM,WASSERSTOFF,WISMUT,WOLFRAM</p> <p>560 DATA XENON,ZINK,ZINN</p> <p>595 REM DATEN ABKUEZUNGEN</p> <p>600 DATA AL,SB,AR,AS,BA,PB,B,BR,CL,CR,FE,F,AU,HE,IR,J,CD,K,CA,CO,C,CU,LI,MO</p> <p>610 DATA MN,MO,NA,NE,NI,P,PT,PU,HG,RA,RN,O,S,SE,AG,SI,N,SR,TH,TI,U,V,H</p> <p>620 DATA BI,W,XE,ZN,SN</p> <p>640 REM DATEN ORDNUNGSZAHLEN</p> <p>650 DATA 13,51,18,33,56,82,5,35,17,24,26,9,79,2,77,53,48,19,20,27,6,29</p> <p>660 DATA 3,12,25,42,11,10,28,13,78,94,80,88,86,8,16,34,47,14,7,38</p> <p>670 DATA 90,22,92,23,1,83,74,54,30,50</p> <p>695 REM DATEN ATOMGEWICHTE</p> <p>700 DATA 26.97,121.76,39.944,74.91,137.36,207.21,10.82,79.919,35.457,52.01</p> <p>710 DATA 55.85,19,197,4.003,192.2,126.92,112.41,39.096,40.08,58.94,12.011</p> <p>720 DATA 63.54,6.94,24.32,54.94,95.95,22.991,20.183,58.71,30.98,195.09</p> <p>730 DATA 242,200,61,226.05,222,16,32.066,78.96,107.88,28.06,14.008</p> <p>740 DATA 87.63,232.05,47.9,238.07,50.95,1.008,209,183.86,131.3,65.38,118.7</p> <p>745 REM DATEN NICHTE</p> <p>750 DATA 2.7,6.67,.0018,5.72,3.5,11.34,1.73,3.14,1.557,7.14,7.86,.0017</p> <p>760 DATA 19.25,.00018,22.42,4.942,8.64,.862,1.545,8.83,3.510,8.933,.534</p> <p>770 DATA 1.75,7.3,10.2,.971,.0009,8.8,1.83,21.4,-,13.55,-,009,.0014,2.07</p> <p>780 DATA 4.46,10.5,2.34,.00125,2.63,11.7,4.43,19.6,07,.0009,9.8,19.3</p> <p>790 DATA .0059,7.14,7.28</p> <p>795 REM DATEN SCHMELZPUNKTE</p> <p>800 DATA 658,630,-190,-,704,327,2400,-7.3,-100,1800,1525,-218,1063,-272</p> <p>810 DATA 2454,114,321,64,851,1490,-,1083,180,650,1221,2622,89</p> <p>820 DATA -248,8,1452,44,1771,-,-39,700,-71,-218,7,113,220,960,1414</p> <p>830 DATA -210,771,1827,1727,1689,1726,-262,271,3380,-111.9,419,232</p> <p>845 REM DATEN SIEDEPUNKTE</p> <p>850 DATA 2270,1440,-186,616,1537,1690,2550,58,-34,2600,2450,-187,2677</p> <p>860 DATA -269,4400,184,770,762,1439,3185,-,2360,1372,1120,2152,4800,880</p> <p>870 DATA -246,3075,280,3800,-,357,1140,-61.9,-182.97,444,688,1930,2630</p> <p>880 DATA -195.8,1360,3530,-,-,3000,-272.78,1560,6000,-108.1,906,2430</p> <p>900 REM DATEN PERIODISCHES SYSTEM</p> <p>910 DATA III,B,M,V,B,O,0,A,M,V,B,N,II,A,P,IV,B,P,III,B,L,VII,B,N</p> <p>920 DATA VII,B,M,VI,A,N,VIII,A,N,VII,B,L,I,B,P,0,A,K,VIII,A,P,VII,B,O</p> <p>930 DATA II,B,O,I,A,N,II,A,N,VIII,A,N,IV,B,L,I,B,N,I,A,L,II,A,M,VII,A,N</p> <p>940 DATA VI,A,O,I,A,M,0,A,L,VIII,A,N,V,B,M,VIII,A,P,-,-,11,B,P,II,A,Q,0,A,P</p> <p>950 DATA VI,B,L,VI,B,M,VI,B,N,I,B,O,IV,B,M,V,B,L,II,A,O,-,-,IV,A,N</p> <p>960 DATA -,-,-,V,A,N,I,A,K,V,B,P,VI,A,P,0,A,O,II,B,N,IV,B,O</p>		

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 5 : Periodensystem	Seite 127
-----------------------	--	--------------

(5.) SCHRITT : Variablenliste

A\$ = Antwortstring (Ja/Nein)

B\$ = Speicherfeld zur Aufnahme des vom Benutzer gewünschten Elements

I = Laufindex

J = Laufindex

N = Zahl der Elemente im Datenbestand

N\$ = Dateielemente

(6.) SCHRITT : Programmbeschreibung

Satz 10-50 : Überschrift

Satz 55 : Angabe "Moment bitte", da das Einlesen des gesamten Datenbestandes einige Sekunden dauert

Satz 60 : Angabe der Elementanzahl (es sind in dieses Programm nicht alle chemischen Elemente aufgenommen, sondern nur 52 Stück. Ggf. muß also Satz 60 verändert werden, falls die DATA-Statements verlängert werden sollten)

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 5 : Periodensystem	Seite 128
	<p>Satz 7φ : Dimensionierung</p> <p>Satz 75-1φ8 : Einlesen aller Informationen</p> <p>Satz 11φ : Anforderung des Elements, über das Informationen gewünscht werden</p> <p>Satz 12φ-13φ : Aufsuchen des gewünschten Elements im Datenbestand; wird es gefunden, weiter bei 16φ; wird es hingegen nicht gefunden, weiter bei 14φ</p> <p>Satz 14φ-15φ : Ist das gewünschte Element nicht im Datenbestand vorhanden, so erfolgt ein entsprechender Ausdruck; danach Sprung zum Satz 29φ</p> <p>Satz 16φ : Löschen des Bildschirms</p> <p>Satz 17φ : Ausgabe des Elementnamens und des chemischen Kürzels in weißer Schrift</p> <p>Satz 18φ : Umschalten auf Hellblau und eine Leerzeile</p> <p>Satz 19φ-28φ : Ausgabe der übrigen Informationen für das gewünschte Element</p> <p>satz 29φ-3φφ : Abfrage, ob eine weitere Auskunft gegeben werden soll, wenn ja, Löschen des Bildschirms und</p>	

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 5 : Periodensystem	Seite 129
-----------------------	--	--------------

Sprung nach 11ø

wenn nein, weiter bei 31ø

Satz 31ø : Beendigung des Programms

Satz 495-96ø : Datenbestand in DATA-Statements

7. SCHRITT : Ergebnisse

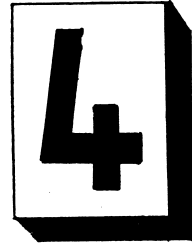
Geben wir z.B. als gewünschtes Element das Element Schwefel ein, so antwortet der Rechner :

SCHWEFEL		S
ORDNUNGSZAHL	:	16
GRUPPE	:	VI
NEBENGRUPPE	:	B
SCHALE	:	M
ATOMGEWICHT	:	32.ø66
WICHTE	:	2.ø7
SCHMELZPUNKT	:	113
SIEDEPUNKT	:	444

EIN ANDERES ELEMENT (J/N) ?

Voß C64/ Schule	Kapitel 3 : Chemie Abschnitt 5 : Periodensystem	Seite 130

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 1 : Vorbemerkung	Seite 131
-----------------------	--	--------------



Kapitel 4 : Physik
=====

4.1 Vorbemerkung

Der Bereich der Schulphysik bietet ähnliche Problemstellungen, wie wir sie auch schon teilweise im Bereich der Mathematik vorgefunden haben (siehe Kapitel 2). Allerdings kommen hier nun häufig einige Besonderheiten hinzu, die es nahelegen, wieder von den Graphikmöglichkeiten des Rechners Gebrauch zu machen.

Deshalb soll - über das hinaus, was im vorangegangenen Kapitel schon über die Methoden der Farb- und Graphikprogrammierung gesagt wurde - in diesem einleitenden Abschnitt noch einmal auf die Graphik-BASIC-Anweisungen etwas näher eingegangen werden.

4.2 Graphik-Programmierung

Bei der Verwendung eines Commodore C 64 werden unter dem Stichwort Graphik-Programmierung die folgenden Aspekte angesprochen :

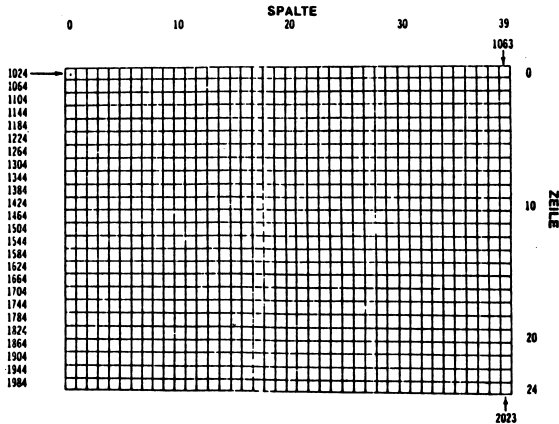
1. Erzeugung der sog. Normalgraphik;
2. Erzeugung von Farben in einem laufenden Programm;
3. Erzeugung der sog. hochauflösenden Graphik.

Den Unterschied zwischen der Normalgraphik und der hochauflösenden Graphik kann man sich verdeutlichen, wenn man die Bildschirmaufteilung vor Augen hat :

Bei der Normalgraphik kann man sich den Bildschirm aufgeteilt denken in 40 Spalten (0 bis 39) und in 25 Zeilen (0 bis 24). Auf dem Bildschirm können also gemäß dieser Aufteilung maximal $40 \cdot 25 = 1000$ Symbole untergebracht werden.

Diese Aufteilung wird schematisch durch die folgende Abbildung verdeutlicht, die gleichzeitig auch die Speicheradressen dieser 1000 Einzelfelder angibt bzw. berechenbar macht.

Abb. 1 : Bildschirmadressen

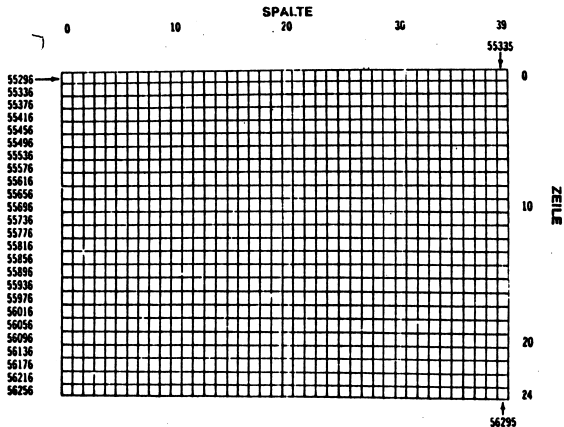


Wie diese Adressen benutzt werden können, um Graphiken anzufertigen, soll weiter unten besprochen werden. Zunächst aber zu den anderen genannten Aspekten :

Zu dem zweiten Aspekt, der Erzeugung von unterschiedlichen Farben während des Programmablaufs, wurde schon im vorangegangenen Kapitel das Wesentliche gesagt.

Hier nun ist ergänzend anzumerken, daß jedes einzelne Bildschirmfeld nicht nur unter Benutzung der oben genannten Adressen einzeln belegt werden kann, sondern es kann auch einzeln farblich unterschiedlich gestaltet werden. Dazu benötigt man die sog. Farbspeicheradressen dieser Bildschirmfelder, die in der folgenden Abbildung genannt werden (bzw. berechenbar werden) :

Abb. 2 : Farbspeicheradressen



Zum dritten Aspekt, der hochauflösenden Graphik ist hier folgendes anzumerken :

Jedes der 1000 Bildschirmfelder, das mit je einem Symbol (z.B. einem Buchstaben) belegt werden kann, baut sich wiederum aus einer Matrix aus 8×8 Punkten auf, die einzeln ansprechbar sind.

Die Ansprache der einzelnen Matrixpunkte ist aber beim Commodore C 64 recht umständlich und erfordert eine Reihe nicht ganz übersichtlicher (wenn auch nicht sehr schwieriger) Rechenoperationen - es sei denn man bedient sich eines der zur Verfügung stehenden Graphikprogramme, die den Benutzer von der Last derartiger Berechnungen befreien.

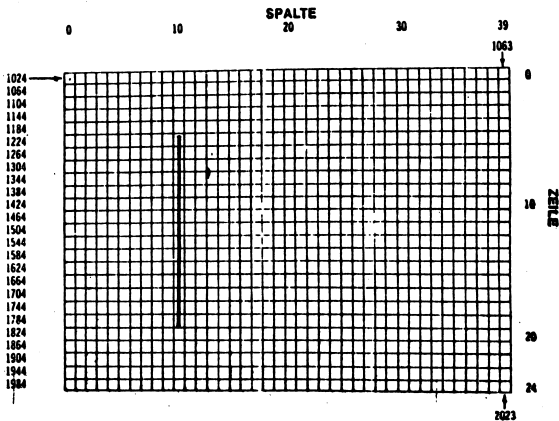
Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 2 : Graphik	Seite 135
<p data-bbox="221 344 947 536">Da dieses Buch aber nun speziell sich auch an den "Computer-Einsteiger" wendet, und weil die hier vorgestellten Programme auch ohne Software-Unterstützung "laufen" sollen, werden wir im folgenden auf die hochauflösende Graphik verzichten und sie erst für spezielle Probleme im letzten Kapitel verwenden.</p> <p data-bbox="221 580 947 676">Wir nehmen dabei in Kauf, daß die graphischen Abbildungen, die in diesem Kapitel 4 erzeugt werden, nicht ganz so elegant aussehen, wie man das vielleicht gern hätte.</p>		

4.3

BASIC-Graphikanweisungen

Nehmen wir an, wir wollten einen senkrechten Strich auf dem Bildschirm zeichnen, wie es die folgende Abbildung schematisch verdeutlicht :

Abb. 3 : Senkrechter Strich



Dies kann schon mit den in Kapitel 3 besprochenen Möglichkeiten erledigt werden. Das entsprechende Programm, das der Leser erproben sollte, könnte folgendermaßen aussehen :

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 3 : BASIC	Seite 137
-----------------------	---	--------------

```

1Ø REM SENKRECHTER STRICH
2Ø FOR I = 1 TO 5 : ? : NEXT I
3Ø FOR I = 1 TO 15
4Ø ? TAB(1Ø) " | "
5Ø NEXT I
6Ø END

```

SHIFT und **B**

Soll dieser Strich in einer speziellen Farbe, z.B. in roter Farbe gezeichnet werden, so ist der Satz 4Ø wie folgt auszuwechsln :

```
4Ø ? TAB(1Ø) " CTRL 3 | "
```

Es leuchtet unmittelbar ein, daß unter Benutzung der auf der C 64-Tastatur vorhandenen Graphiksymbole auch andere "Bausteine" für einen derartigen Strich verwendet werden können.

Der Leser erprobe deshalb das obige Programm auch mit folgender Version des Satzes 4Ø :

```
4Ø ? TAB(1Ø) " Ø " SHIFT und Q
```

Sicherlich kann der Leser nach dieser Vorübung sich ausdenken, wie ein Programm aussehen müßte, das einen waagrechten anstelle eines senkrechten Strichs zieht.

Unter Benutzung der oben vorgestellten Bildschirm-
speichereinteilung lassen sich solche Zeichenprogramme
aber auch auf andere Weise erstellen :

Dazu benötigt man ergänzend die folgende BASIC-Anwei-
sung :

Statement 15:

nn POKE Speicheradresse,Zahl

In dieser Anweisung kann an der Stelle "Zahl" ein Wert
zwischen 0 und 255 stehen. Die Symbole, die mit diesen
Zahlenwerten darstellbar sind, werden in der folgenden
Tabelle dargestellt : BILDSCHIRM CODES

Satz 1	Satz 2	Poke	Satz 1	Satz 2	Poke	Satz 1	Satz 2	Poke
@		0	V	v	22	.		44
A	a	1	W	w	23	-		45
B	b	2	X	x	24	.		46
C	c	3	Y	y	25	/		47
D	d	4	Z	z	26	0		48
E	e	5	[27	1		49
F	f	6	£		28	2		50
G	g	7]		29	3		51
H	h	8	↑		30	4		52
I	i	9	←		31	5		53
J	j	10	SPACE		32	6		54
K	k	11			33	7		55
L	l	12	"		34	8		56
M	m	13	#		35	9		57
N	n	14	\$		36	:		58
O	o	15	%		37	:		59
P	p	16	&		38	<		60
Q	q	17	'		39	=		61
R	r	18	(40	>		62
S	s	19)		41	?		63
T	t	20	*		42			
U	u	21	+		43			

Satz 1	Satz 2	Poke	Satz 1	Satz 2	Poke	Satz 1	Satz 2	Poke
		64		V	86			108
	A	65		W	87			109
	B	66		X	88			110
	C	67		Y	89			111
	D	68		Z	90			112
	E	69			91			113
	F	70			92			114
	G	71			93			115
	H	72			94			116
	I	73			95			117
	J	74	SPACE		96			118
	K	75			97			119
	L	76			98			120
	M	77			99			121
	N	78			100			122
	O	79			101			123
	P	80			102			124
	Q	81			103			125
	R	82			104			126
	S	83			105			127
	T	84			106			
	U	85			107			

Die Codes 128-255 ergeben die invers dargestellten Zeichen der Codes 0-127.

An den freien Stellen in Spalte 2 stimmen die beiden Sätze überein.

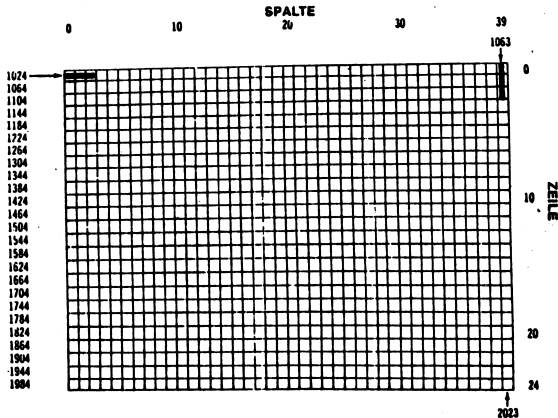
Anhand dieser Übersicht kann der Leser herausfinden, was das folgende Programm tut :

```

1Ø POKE 1Ø24,64
2Ø POKE 1Ø25,64
3Ø POKE 1Ø26,64
4Ø POKE 1Ø63,66
5Ø POKE 11Ø3,66
6Ø POKE 1143,66
7Ø END

```

Offenbar erzeugt dieses Programm das folgende Bild :



Wie findet man aufgrund der Bildschirmspeichertabelle von Abb. 1 eine beliebige Speicheradresse ?

Nehmen wir an, wir wollten einen "Ball" in die 15. Spalte und 10. Zeile des Bildschirms per POKE-Anweisung zeichnen.

Die Code-Zahl für den "Ball" ("Zahl" in der POKE-Anweisung) ist der Wert 81 (siehe obige Tabelle).

Die Speicheradresse des Feldes, das uns interessiert, das Feld nämlich an der "Kreuzung" der 15. Spalte mit der 10. Zeile findet man folgendermaßen (vergl. dazu auch Abb. 1) :

$$\text{Adresse} = 1024 + 15 + 10 * 40 = 1439$$

↑
↑
↑

Startadresse
Spalte
Zeile

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 3 : BASIC	Seite 141
-----------------------	---	--------------

Die entsprechende Anweisung lautet also :

nn POKE 1439,81

Wenn wir den mit dieser Anweisung erzeugten "Ball" in einer anderen Farbe sehen wollen, so gilt die folgende Überlegung :

An der gleichen Speicherstelle, an der der "Ball" gezeichnet wurde, kann man für eine andere Farbgebung sorgen. Dies ist unter Benutzung der Farbspeicheradressen möglich, die in Abb. 2 vorgestellt wurden.

Entsprechend wie oben die Bildschirmadresse ausgerechnet wurde, läßt sich nun auch die Farbspeicheradresse ausrechnen :

$$\text{Farbadresse} = 55296 + 15 + 1\phi * 4\phi = 55711$$

Die Anweisung zur Farbgebung lautet also entsprechend :

nn POKE Farbadresse,Zahl

Hierbei ist "Zahl" ein Wert zwischen ϕ und 15, der die Farben kennzeichnet gemäß der Tabelle, die in Abschnitt 3.1 schon vorgestellt wurde.

Um nun also unseren "Ball" rot zu färben, benötigen wir das folgende kleine Programm :

```
1Ø POKE 55711,2
2Ø POKE 1439,81
3Ø END
```

Farbgebung
"Ball"

Wie sieht nun ein entsprechendes Programm für einen waagrechten Strich, sagen wir in Zeile 1Ø bei Spalte 5 beginnend und bei Spalte 3Ø endend, aus ?

Die Adresse des ersten zu belegenden Feldes lautet :

$$1Ø24 + 5 + 1Ø * 4Ø = 1429$$

Danach folgen die Adressen 143Ø bis 1454.

Die entsprechenden Farbspeicheradressen lauten :

$$55296 + 5 + 1Ø * 4Ø = 557Ø1$$

bis

$$55296 + 3Ø + 1Ø * 4Ø = 55726$$

Der Unterschied zwischen Bildschirmadresse und Farbspeicheradresse beträgt also immer 54272 (diese Zahl sollte man sich merken, wenn man oft in dieser Weise programmiert).

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 3 : BASIC	Seite 143
-----------------------	---	--------------

Das Programm lautet demnach :

```

1Ø ?CHR$(147)
2Ø FOR I = 1429 TO 1454
3Ø J = I+54272
4Ø POKE J,2
5Ø POKE I,64
6Ø NEXT I
7Ø END

```

Wollen wir einen senkrechten Strich zeichnen, so ist zu berücksichtigen, daß untereinanderliegende Bildschirmfenster einen Adreßwertunterschied von 4Ø aufweisen.

Somit taugt das folgende Programm für einen senkrechten Strich :

```

1Ø ?CHR$(147)
2Ø FOR I = 1434 TO 1834 STEP 4Ø
3Ø POKE I,66
4Ø NEXT I
5Ø END

```

Verändert man die Schrittweite, so können auch schräge Striche gezeichnet werden :

Erhöhen wir beispielsweise die Schrittweite auf 41, so ist in einer POKE-Anweisung innerhalb einer Programmschleife, die jeweils folgende Adresse in der nächsten Bildschirmzeile um ein Feld nach rechts versetzt. Entsprechend erhalten wir eine Versetzung nach links, würden wir mit der Schrittweite 39 arbeiten.

4.4

Die Federwaage

①. SCHRITT : Vorstellung des Problems

Wenn man Werkstücke unterschiedlicher Gewichte an eine Federwaage hängt, so wird diese unterschiedlich weit ausgedehnt, d.h. sie reagiert in unterschiedlicher Weise auf die verschiedenen Gewichtskräfte.

Es gilt dabei ein Proportionalitätsgesetz, das besagt, daß z.B. bei Verdopplung der Gewichtskraft auch die erzielte Verlängerung der Federwaage sich verdoppelt.

Es gibt also eine sog. Federkonstante, nämlich den Quotienten aus Gewichtskraft F und Federverlängerung s , d.h. es gilt :

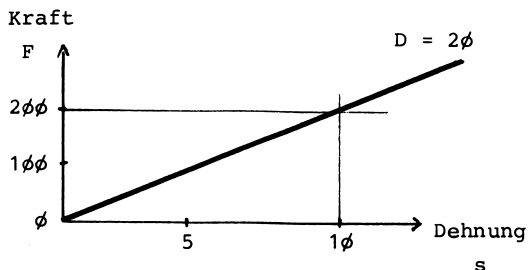
$$D = F/s = \text{konstant}$$

Wenn beispielsweise $D = 2\phi$ ist, heißt dies, daß eine Gewichtskraft von $1\phi\phi$ (cN) zu einer Federverlängerung von 5 cm, eine Gewichtskraft von $2\phi\phi$ (cN) zu einer Federverlängerung von 1ϕ cm führt etc.

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 4 : Federwaage	Seite 145
-----------------------	--	--------------

(2.) SCHRITT : Problemanalyse

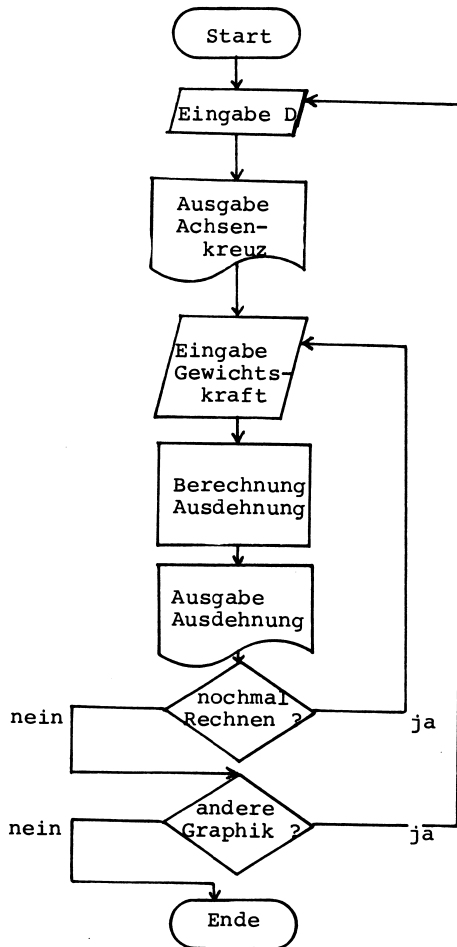
Der oben geschilderte Zusammenhang läßt sich (z.B. für $D = 2\phi$) folgendermaßen graphisch darstellen :



Das hier zu entwickelnde BASIC-Programm soll für alternative Werte der Federkonstanten D das entsprechende graphische Bild auf dem Bildschirm ausgeben, so daß also die folgenden Programmteile erforderlich werden :

- Erzeugung eines Achsenkreuzes,
- Eingabe eines alternativen D-Wertes,
- Ausgabe der Konstanzlinie (siehe obige Abbildung),
- Beantwortung der Frage nach der Ausdehnung der Federwaage bei Eingabe einer beliebigen Gewichtskraft.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 4 : Federwaage	Seite 147
-----------------------	--	--------------

4. SCHRITT : Programm

```

10 REM P1-FEDERWAAGE
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR GRAPHISCHEN VERANSCHAULI-"
40 PRINTTAB(3)"CHUNG DES FEDERWAGEN-GESETZES."
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984":PRINT
55 PRINT:PRINT:PRINT:PRINT
60 INPUT"WERT DER FEDERKONSTANTEN D : ";D
70 PRINTCHR$(147):REM ACHSENKREUZ
75 PRINT:PRINT:PRINTTAB(3)"KRAFT";
77 PRINTTAB(15)"FEDERKONSTANTE D = ";D
80 FOR I=1229 TO 1829 STEP 40
90 POKE I,66:NEXT I
100 FOR I=1829 TO 1859
110 POKE I,64:NEXT I
115 POKE 1898,3:POKE 1899,13:REM TEXT CM
120 REM GRENZPUNKTE
130 S0=5:Z0=20
140 S1=30:Z1=5
150 REM GRAPHIK
170 Z=(200/6)/D
180 B=D/(200/6):B=-B
190 FOR S=S0 TO S1
200 Z=(20+D/6.7)+B*S
210 Z=INT(Z+.5)
220 A=1024+S+Z*40
230 IF A<1184 OR A >1900 THEN 260
240 POKE A+54272,2
250 POKE A,81
255 NEXT S
260 POKE A+2,4:REM TEXT D
263 FOR I=1TO9999:NEXT I
265 PRINTCHR$(147)
270 PRINT:INPUT"KRAFT F          = ";F
280 FW=F/D
290 PRINT:PRINT:PRINT "FEDERKONSTANTE = ";D
300 PRINT:PRINT:PRINT "FEDERWEG          = ";FW
310 PRINT:PRINT:PRINT:INPUT"NOCH EINE RECHNUNG (J/N) : ";A$
320 IF A$="J" THEN 265
330 PRINT:PRINT:INPUT"ANDERE GRAPHIK          (J/N) : ";A$
340 IF A$="J" THEN PRINT CHR$(147):GOTO 60
350 PRINT:PRINT"ENDE":END

```

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 4 : Federwaage	Seite 148
-----------------------	--	--------------

(5) SCHRITT : Variablenliste

A = Bildschirmadresse

A\$ = Antwortstring (Ja/Nein)

B = $-1/Z$ (Größe zur Normierung der Bildschirmausgabe derart, daß der Bildschirm voll ausgenutzt wird)

D = Federkonstante

F = Kraft

FW = Federweg

I = Laufindex

S = Spaltenzählindex

S \emptyset = Anfangsspalte beim Achsenkreuz

S1 = Endspalte beim Achsenkreuz

Z = Normierte Zeilenbestimmung für die zu zeichnende Konstanlinie

Z \emptyset = Anfangszeile beim Achsenkreuz

Z1 = Endzeile beim Achsenkreuz

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 4 : Federwaage	Seite 149
-----------------------	--	--------------

(6.) SCHRITT : Programmbeschreibung

- Satz 1 ϕ -55 : Überschrift
- Satz 6 ϕ : Eingabe der Federkonstanten D
- Satz 7 ϕ -115 : Zeichnen des Achsenkreuzes (senkrechte Achse : 8 ϕ -9 ϕ ; waagrechte Achse : 10 ϕ -11 ϕ ; Achsenbezeichnungen : 75,77,26 ϕ)
- Satz 12 ϕ -15 ϕ : Setzen der Graphik-Grenzpunkte
- Satz 17 ϕ -18 ϕ : Normierung zum Zwecke der geschickten Bildschirmausnutzung
- Satz 19 ϕ -255 : Zeichnen der Konstanzfunktion :
- 20 ϕ -21 ϕ : Bildschirmzeile
 - 22 ϕ : Adresse
 - 23 ϕ : Kontrolle der Adresse, damit der zulässige Bereich nicht verlassen wird
 - 24 ϕ : Zeichnen in roter Farbe
 - 25 ϕ : Zeichnen
- Satz 263-265 : Warteschleife und Löschen des Bildschirms
- Satz 27 ϕ : Anforderung einer Kraft F
- Satz 28 ϕ : Berechnung des Federweges

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 4 : Federwaage	Seite 150
-----------------------	--	--------------

Satz 29 ϕ -3 $\phi\phi$: Ausgabe der Ergebnisse

Satz 31 ϕ : Abfrage, ob noch eine Rechnung;
wenn ja, zurück nach 265
wenn nein, weiter bei 33 ϕ

Satz 33 ϕ : Abfrage, ob eine andere Graphik
wenn ja, zurück nach Satz 6 ϕ nach Löschen des Bildschirms
wenn nein, weiter bei Satz 35 ϕ

Satz 35 ϕ : Ende des Programms

7. SCHRITT : Ergebnisse

Nach der Eingabe der Inputinformation erzeugt das Programm eine Graphik, wie sie schematisch auf einer der vorangegangenen Seiten schon vorgestellt wurde.

Nach einer Pause fordert dann das Programm eine Kraft F an und berechnet den dazugehörigen Federweg. Ein Rechenbeispiel dazu dürfte sich an dieser Stelle erübrigen (vergl. dazu 1. Schritt, wo sich ein entsprechendes Beispiel ja findet).

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 5 : Archimedes	Seite 151
-----------------------	--	--------------

4.5

Der Satz des Archimedes

①, SCHRITT : Vorstellung des Problems,

Der Satz des Archimedes lautet :

Der Auftrieb, den ein in eine Flüssigkeit getauchter Körper erfährt, ist gleich dem Gewicht der verdrängten Flüssigkeit.

Dieser Satz beschreibt die bekannte Erfahrung, daß es zunehmender Kraft bedarf, um zum Beispiel einen leeren Eimer mit dem Boden nach unten unter Wasser zu drücken.

Ein immer tiefer in eine Flüssigkeit eintauchender Körper erfährt einen zunehmenden scheinbaren Gewichtsverlust, der durch den wachsenden Auftrieb vorgetäuscht wird.

Es soll nun ein Programm vorgestellt werden, welches auf der Grundlage dieser Beziehungen für einen unbekanntes Körper, der in eine bekannte Flüssigkeit eingetaucht wird (z.B. in Wasser), Volumen und Wichte (spezifisches Gewicht) bestimmt, nachdem sein Gewicht an der Luft bekannt ist.

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 5 : Archimedes	Seite 152
-----------------------	--	--------------

②. SCHRITT : Problemanalyse

Wie bei jeder konkreten Problemstellung, ist auch hier nach dem folgenden bewährten Muster zu verfahren :

E	= Eingabe
V	= Verarbeitung
A	= Ausgabe

Das heißt, wir überlegen zunächst, welche Eingabeinformationen der Rechner benötigt, wenn er das oben formulierte Problem lösen soll. Er benötigt folgendes :

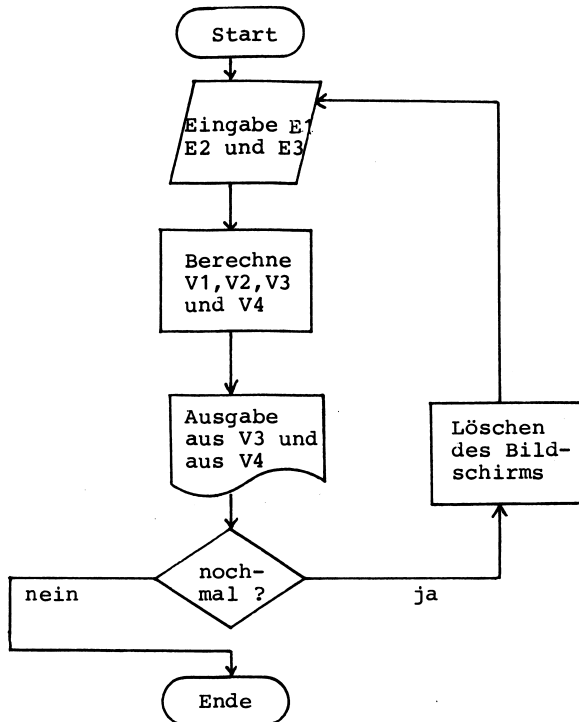
- E1. Gewicht des unbekanntes Körpers in der Luft;
- E2. Scheinbares Gewicht des Körpers in der Flüssigkeit, in die er eingetaucht wird;
- E3. Wichte der Flüssigkeit, die zur Überprüfung verwendet wird.

Wenn diese Informationen vorliegen, dann kann im Verarbeitungsschritt bestimmt werden :

- V1. Auftrieb und Gewicht der verdrängten Flüssigkeitsmenge;
- V2. Volumen dieser Flüssigkeitsmenge;
- V3. Volumen des verdrängenden, zu prüfenden Körpers;
- V4. Wichte des zu prüfenden, eingetauchten Körpers.

Als Ausgabe sind schließlich die in den letzten Punkten 3. und 4. erzielten Ergebnisse vorzusehen.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 5 : Archimedes	Seite 154
-----------------------	--	--------------

④ SCHRITT : Programm

```

10 REM P2-ARCHIMEDES
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR VERWENDUNG DES SATZES DES"
40 PRINTTAB(14)"ARCHIMEDES.":PRINT:PRINT
50 PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
55 PRINT:PRINT"MIT DEM SATZ DES ARCHIMEDES KOENNEN VO-"
56 PRINT"LUMEN UND WICHTE EINES UNBEKANNTEN"
57 PRINT"KOERPERS BESTIMMT WERDEN.":PRINT:PRINT
70 PRINT"EINZUGEBEN SIND DIE FOLGENDEN INFORMA-"
80 PRINT"TIONEN : ":PRINT
90 PRINTTAB(5)"- GEWICHT DES KOERPERS AN DER LUFT"
100 PRINTTAB(5)" (GL)"
110 PRINT:PRINTTAB(5)"- SCHEINBARES GEWICHT DES KOERPERS"
120 PRINTTAB(5)" IN DER FLUESSIGKEIT (GS)"
130 PRINT:PRINTTAB(5)"- WICHTE DER FLUESSIGKEIT (WF)"
140 PRINT:PRINT:INPUT"GL = ";GL
150 PRINT: INPUT"GS = ";GS
160 PRINT: INPUT"WF = ";WF
170 GV=GL-GS
175 VV=GV/WF
180 VK=VV
190 WK=GL/VK
200 PRINTCHR$(147)
210 PRINT"ZU PRUEFENDER KOERPER : ":PRINT
220 PRINT:PRINT" VOLUMEN = ";VK
230 PRINT:PRINT" WICHTE = ";WK
240 PRINT:PRINT:PRINT:INPUT"NOCHMAL ? (J/N) ";A$
250 IF A$="J" THEN PRINTCHR$(147):GOTO 70
260 PRINT:PRINT"ENDE DER BERECHNUNGEN":END

```

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 5 : Archimedes	Seite 155
-----------------------	--	--------------

5. SCHRITT : Variablenliste

A\$ = Stringvariable für die Antworten Ja/Nein.

GL = Gewicht des Körpers an der Luft

GS = Scheinbares Gewicht des Körpers in der Flüssigkeit

GV = Scheinbarer Gewichtsverlust

VK = Volumen des Körpers

VV = Volumen der verdrängten Flüssigkeit

WF = Wichte der Flüssigkeit

WK = Wichte des Körpers

6. SCHRITT : Programmbeschreibung

Satz 10-13 : Überschrift und erläuternde Informationen

Satz 14-16 : Anforderung der Input-Informationen

Satz 17-19 : Verarbeitungsschritte (Berechnungen)

Satz 20-23 : Ausgabe der Ergebnisse

Satz 24-26 : Abfrage, ob weitere Berechnung (zurück nach 7), bzw. Beendigung des Programms

7. SCHRITT : Ergebnisse

Geben wir zum Beispiel die folgenden Input-Informationen :

E1: Gewicht des Körpers = 70 g;

E2: Scheinbares Gewicht des Körpers in der Prüfflüssigkeit (zu messen mit einer Federwaage) = 40 g;

E3: Wichte der Prüfflüssigkeit = 1 (Wasser),

dann erzeugt das Programm die folgenden Ergebnisse :

ZU PRUEFENDER KOERPER :

VOLUMEN = 30

WICHTE = 2.3333333

NOCHMAL (J/N) ?

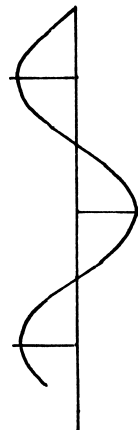
4.6 Pendelbewegung

1. SCHRITT : Vorstellung des Problems

Die Pendelbewegung ist ein wichtiger physikalischer Grundtatbestand, der in der Lehre vom Schall (Akustik), in der Lehre vom Licht (Optik) und in der Elektrizitätslehre eine wichtige Rolle spielt.

Pendelbewegungen folgen einer Sinusschwingung, wie es die nebenstehende Skizze zeigt.

Es soll nun ein Programm vorgestellt werden, welches für alternative Amplituden (A) und für alternative Periodenlängen (P), die ihrerseits die sog. Frequenz (F) einer Schwingung ausmachen, eine solche Sinusschwingung graphisch veranschaulicht.



Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 6 : Pendel	Seite 153
-----------------------	--	--------------

②. SCHRITT : Problemanalyse

Für das geplante Programm müssen wir als Input-Informationen die Amplitude der Schwingung und ihre Periodenlänge vorgeben.

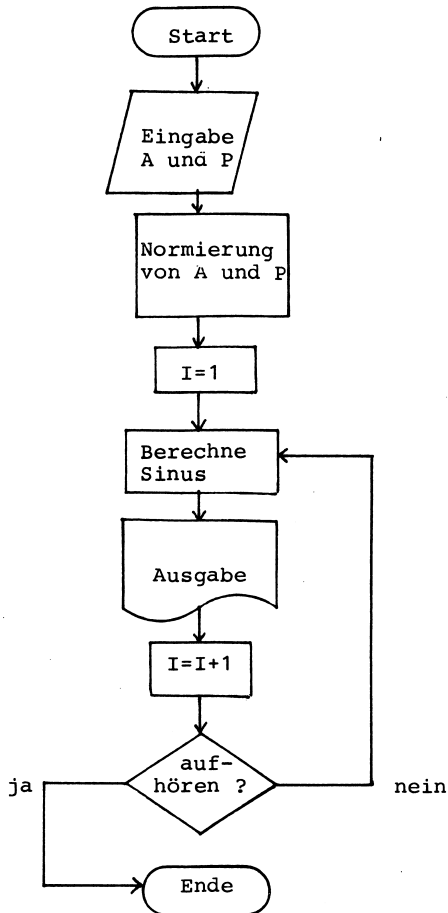
Das Programm kann dann veranlassen, daß - wie in obiger Skizze - die Sinusschwingung von oben nach unten über den Bildschirm "läuft" (was zu einem einfacheren Programm führt, als wenn diese Schwingung horizontal ausgegeben werden sollte).

Wesentlicher Inhalt des Programms muß es dabei sein, Amplitude und Periodenlänge so zu normieren, daß der Bildschirm zur Darstellung der Schwingung optisch gut ausgenutzt wird.

Dies kann erreicht werden, indem wir die Mittellinie der Schwingung in die Spalte 20 legen und die Amplitude, also den größtmöglichen Ausschlag z.B. auf den Maximalwert 19 ausrichten.

Dieses Programm kann dann wieder als anschauliches Beispiel dafür dienen, wie man die Normalgraphik-Eigenschaften des Commodore C 64 geschickt benutzen kann. Wir werden dann im letzten Kapitel sehen, wie man die gleiche Frage unter Benutzung der hochauflösenden Graphik bearbeiten kann (siehe auch Abschnitt 4.1).

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 6 : Pendel	Seite 160
-----------------------	--	--------------

4. SCHRITT : Programm

```

10 REM P3-PENDEL
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR SIMULATION EINER PENDELBE-"
40 PRINTTAB(16)"WEGUNG.":PRINT:PRINT
50 PRINTTAB(9)"PROF.DR.W.VOSS, 1984":PRINT:PRINT:PRINT
55 PRINT:PRINT"ZUM ANHALTEN RUN/STOP-TASTE DRUECKEN":PRINT
60 INPUT"AMPLITUDE DER SCHWINGUNG : ";A
70 PRINT:INPUT"PERIODENLAENGE           : ";P
75 IF A>19 THEN A=19
80 PRINTCHR$(147)
90 I=1
100 J=(2*PI/P)*I
110 Y=SIN(J)
120 S=20+Y*A
130 FOR K=1 TO S
140 PRINT"●";
150 NEXT K
160 PRINT
170 I=I+1:GOTO 100

```

5. SCHRITT : Variablenliste

- A = Amplitude der Schwingung
- I = Laufindex
- J = Transformierte Größe : $J = (2\pi/p) \cdot I$
(J dient der Normierung der Periodenlänge)
- K = Laufindex, der die Länge der pro Bildschirmzeile auszugebenden Punktstäbe mit Hilfe der Größe S bestimmt
- P = Periodenlänge der Schwingung
- S = Bildschirmspeicherstelle, an der die jeweilige Bildschirmzeile endet
- Y = SIN(J)

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 6 : Pendel	Seite 161
-----------------------	--	--------------

6. SCHRITT : Programmbeschreibung

- Satz 1 ϕ -55 : Überschrift und Erläuterungen
- Satz 6 ϕ -7 ϕ : Anforderung der Input-Informationen
- Satz 75 : Beschränkung der Schwingungsamplitude auf den Maximalwert 19
- Satz 8 ϕ : Löschen des Bildschirms
- Satz 9 ϕ : Beginn Ausdrucksschleife (I = Zeilenzählindex)
- Satz 1 $\phi\phi$: Normierung von I nach Maßgabe der Periodenlänge P zur Hilfsgröße J
- Satz 11 ϕ : Bestimmung des dazugehörigen Sinuswerts
- Satz 12 ϕ : Bestimmung der Bildschirmzeilen-Position des jeweiligen Sinuswertes unter Berücksichtigung der Tatsache, daß der Nullwert in der 2 ϕ .Spalte liegen soll
- Satz 13 ϕ -15 ϕ : Schleife zum Auffüllen der Zeile bis zum jeweiligen Sinuswert mit "Kügelchen"
- Satz 16 ϕ -17 ϕ : Erzwingen eines Zeilenvorschubs und Ausgabe der nächsten Zeile durch Rücksprung zu Satz 1 $\phi\phi$ nach Erhöhung von I um 1

VoB C64/ Schule	Kapitel 4 : Physik Abschnitt 6 : Pendel	Seite 162
-----------------------	--	--------------

7. SCHRITT : Ergebnisse

Die durch dieses Programm erzeugten Ergebnisse entsprechen der Skizze, die im ersten Arbeitsschritt vorgestellt wurde, mit dem Unterschied, daß die Bereiche zwischen dem linken Bildschirmrand und der Sinuslinie mit hellblauen "Kügelchen" gefüllt werden.

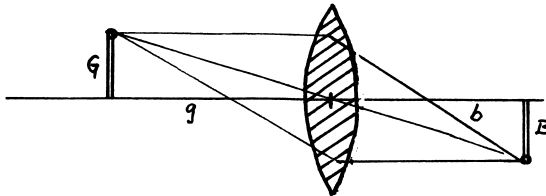
Der Leser kann sich leicht selbst ausdenken, wie das Programm verändert werden müßte, wenn man dieses "Auf-füllen" unterdrücken wollte, also nur diejenigen Positionen z.B. durch "Kügelchen" optisch markiert, die tatsächlich auf der Sinuslinie liegen.

Das hier vorgestellte Programm ist ein Endlosprogramm, das nur durch Betätigung der RUN/STOP -Taste abgebrochen werden kann. Auch dies ist aber leicht abänderbar.

4.7 Optische Abbildung

1. SCHRITT : Vorstellung des Problems

Mit einer konvexen Linse kann - so wie es die folgende Abbildung schematisch zeigt - das Abbild eines beliebigen Gegenstandes erzeugt werden :



Nach diesem Modell funktioniert der Photoapparat aber auch z.B. das menschliche Auge.

Es soll nun ein Programm vorgestellt werden, welches bei vorgegebener Größe des Gegenstandes, Entfernung des Gegenstandes und Brennweite der konvexen Linse die Bildgröße bzw. den Abbildungsmaßstab bestimmt.

2. SCHRITT : Problemanalyse

Das zu erstellende Programm benötigt an Inputinformationen :

1. Gegenstandsgröße,
2. Gegenstands Entfernung,
3. Brennweite der Linse.

Zur Lösung des Problems benötigen wir die sog. Linsengleichung :

$$1/f = 1/g + 1/b$$

wobei :
f = Brennweite
g = Gegenstandsweite
b = Bildweite

Löst man diese Beziehung nach b auf, so erhält man :

$$b = \frac{1}{1/f - 1/g}$$

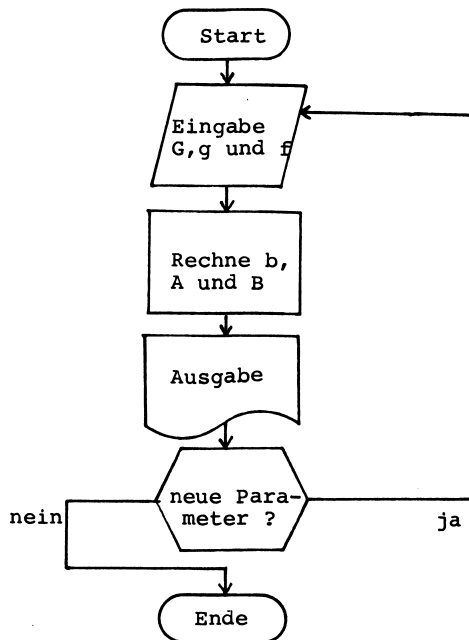
Weiterhin gilt, daß der Abbildungsmaßstab A sich ergibt zu :

$$A = B/G = b/g$$

Wenn nun beispielsweise der abzubildende Gegenstand die Größe $G = 2\phi$ cm hat, so ist mit Hilfe der obigen Beziehungen auch die Bildgröße bestimmbar, nämlich :

$$B = (b \cdot G) / g, \text{ weil } B/G = b/g$$

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 7 : Abbildung	Seite 166
-----------------------	---	--------------

4. SCHRITT : Programm

```

10 REM P4-LINSE
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR BERECHNUNG VON BILDWEITE,"
40 PRINT"BILDGROESSE UND ABBILDUNGSMASSSTAB BEI"
50 PRINTTAB(9)"EINER KONVEXEN LINSE."
60 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984":PRINT:PRINT
70 PRINT"EINZUGEBEN SIND DIE FOLGENDEN DATEN :":PRINT
80 PRINTTAB(5)"- GEGENSTANDSGROESSE : ":INPUT G
90 PRINT:PRINTTAB(5)"- GEGENSTANDSWEITE : ":INPUT GW
100 PRINT:PRINTTAB(5)"- BRENNWEITE/LINSE : ":INPUT F
110 PRINTCHR$(147)
120 BW=1/(1/F-1/GW)
130 A=BW/GW
140 B=(G*BW)/GW
145 BW=INT(BW*100+.5)/100:A=INT(A*100+.5)/100:B=INT(B*100+.5)/100
150 PRINT"AUSGANGSDATEN :":PRINT
160 PRINTTAB(5)"BRENNWEITE : ":F
170 PRINT:PRINTTAB(5)"GEGENSTANDSGROESSE : ":G
180 PRINT:PRINTTAB(5)"GEGENSTANDSWEITE : ":GW
190 PRINT:PRINT:PRINT"ERGEBNISSE : ":PRINT
200 PRINT:PRINTTAB(5)"ABBILDUNGSMASSSTAB : ":A
210 PRINT:PRINTTAB(5)"BILDGROESSE : ":B
220 PRINT:PRINTTAB(5)"BILDWEITE : ":BW
230 PRINT:PRINT:PRINT:INPUT"NOCH EINE RECHNUNG ? (J/N) ":A$
240 IF A$="J" THEN PRINTCHR$(147):GOTO 70
250 PRINT:PRINT"ENDE DER BERECHNUNGEN":END

```

5. SCHRITT : Variablenliste

A = Abbildungsmaßstab
A\$ = Stringvariable (Ja/Nein)
B = Bildgröße
BW = Bildweite
F = Brennweite
G = Gegenstandsgröße
GW = Gegenstandsweite

VoB C64/ Schule	Kapitel 4 : Physik Abschnitt 7 : Abbildung	Seite 167
-----------------------	---	--------------

⑥ SCHRITT : Programmbeschreibung

- Satz 1 ϕ -7 ϕ : Überschrift und Erläuterungen
- Satz 8 ϕ -1 $\phi\phi$: Anforderung der Inputinformationen
- Satz 11 ϕ -145 : Löschen des Bildschirms und Berechnungen.
- Satz 15 ϕ -22 ϕ : Ausgabe der Ergebnisse
- Satz 23 ϕ -24 ϕ : Abfrage, ob weitere Berechnung gewünscht (dann zurück nach Satz 7 ϕ nach Löschen des Bildschirms)
- Satz 25 ϕ : Beendigung des Programms

⑦ SCHRITT : Ergebnisse

Geben wir z.B. ein : $G = 2\phi$ cm, $GW = 3\phi\phi$ cm, $F = 15$ cm,
so erhalten wir :

BRENNWEITE	: 15
GEGENSTANDSGROESSE	: 2 ϕ
GEGENSTANDSWEITE	: 3 $\phi\phi$
ERGEBNISSE :	
ABBILDUNGSMASSSTAB	: . ϕ 5
BILDGROESSE	: 1. ϕ 5
BILDWEITE	: 15.79

NOCH EINE RECHNUNG ? (J/N) ?

4.8

Das Spektrum

①. SCHRITT : Vorstellung des Problems

Das Sonnenlicht, so wie wir es wahrnehmen, ist eine Mischung unterschiedlicher Schwingungsfrequenzen, die nichts anderes als unterschiedliche Lichtfarben sind.

Schickt man Sonnenlicht durch ein Prisma, so werden die unterschiedlichen Lichtfrequenzen, die dieses Licht erzeugen, unterschiedlich stark abgelenkt. Auf einem Papierschirm hinter dem Prisma erscheint deshalb dann ein "Regenbogen"-Farbband, das man als Sonnenlicht-Spektrum bezeichnet.

In dem folgenden Programm soll dieser Vorgang schematisch verdeutlicht werden, wobei aber auf physikalische und mathematische Präzision in diesem Zusammenhang keinen Wert gelegt wird.

Insoweit ist dieses Programm, wie auch schon einige der vorhergehenden, eher als "Demonstrationsprogramm" und weniger als Problemlösungsprogramm zu verstehen. Die Problemlösung besteht hier allenfalls in der Beantwortung der Frage, wie man eine hübsche Farbgraphik programmiert.

Aber auch derartige Fragestellungen sollen ja hier nicht vernachlässigt werden.

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 8 : Spektrum	Seite 169
-----------------------	--	--------------

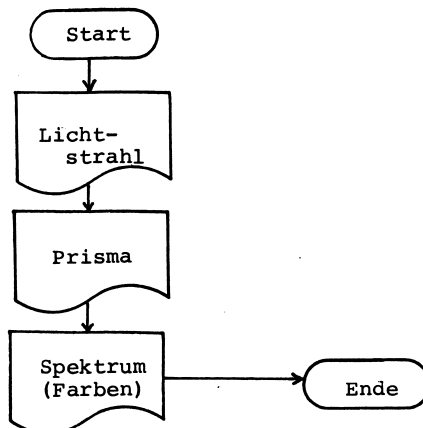
②. SCHRITT : Problemanalyse

Die Problemanalyse ist hier wieder recht einfach :

Das zu entwickelnde Programm besteht im Prinzip nur aus einem Ausgabeteil, in dem die Farbmöglichkeiten des C 64 genutzt werden, um ein Farbband zu erzeugen, welches einem Spektrum ähneln soll.

Nun zeichnet sich natürlich das Regenbogen-Spektrum durch fließende Farbübergänge aus. Genau dies ist aber in einem BASIC-Programm mit den vorhandenen Möglichkeiten nicht darstellbar. Wir können nur unterschiedliche Farben nebeneinander setzen und erreichen so nur eine sehr eingeschränkte Ähnlichkeit mit dem tatsächlichen Spektrum.

③. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM P5-SPEKTRUM
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR DARSTELLUNG EINES SPEKTRUMS"
40 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984":PRINT
50 FOR I=1 TO 4000:NEXT I
60 PRINTCHR$(147)
70 D=54272
100 POKE 1281,78:POKE 1320,78
110 POKE 1360,67:POKE 1361,67:POKE 1362,67
120 POKE 1363,67
130 POKE 1323,77:POKE 1282,77
135 FOR I=1TO1000:NEXTI
140 FOR S=1269 TO 1280
150 POKE S,67:FORI=1TO40:NEXTI:NEXT S
160 FOR S=1324 TO1684 STEP 42
170 POKE S,46:FOR I=1TO40:NEXT I
175 FORI=1TO40:NEXTI
180 NEXT S
190 FOR S=1283 TO 1420 STEP 45
200 POKE S,46:FOR I=1 TO 40:NEXT I
210 NEXT S
230 FOR I=1422 TO 1702 STEP 40
232 POKE I+D,0:POKE I,81:NEXT I
236 FORI=1TO99:NEXTI
240 POKE 1701+D,0:POKE 1701,81
245 FORI=1TO99:NEXTI
250 POKE 1661+D,4:POKE 1661,81
255 FORI=1TO99:NEXTI
260 POKE 1621+D,14:POKE 1621,81
265 FORI=1TO99:NEXTI
270 POKE 1581+D,5:POKE 1581,81
275 FORI=1TO99:NEXTI
280 POKE 1541+D,7:POKE 1541,81
285 FORI=1TO99:NEXTI
290 POKE 1501+D,8:POKE 1501,81
295 FORI=1TO99:NEXTI
300 POKE 1461+D,2:POKE 1461,81
305 FORI=1TO99:NEXTI
310 POKE 1421+D,0:POKE 1421,81
320 FOR I=1TO8000:NEXTI
330 PRINTCHR$(147):PRINT"ENDE"
340 PRINT:PRINT"NEUSTART NUR MIT RUN":END
```

VoB C64/ Schule	Kapitel 4 : Physik Abschnitt 8 : Spektrum	Seite 171
-----------------------	--	--------------

(5.) SCHRITT : Variablenliste

D = Adressunterschied zwischen Bildschirm- und Farbspeicheradresse.
I = Laufindex
S = Speicheradresse

(6.) SCHRITT : Programmbeschreibung

Satz 1 ϕ -4 ϕ : Überschrift.

Satz 5 ϕ : Warteschleife

Satz 6 ϕ : Löschen des Bildschirms

Satz 7 ϕ : Belegen der Variablen D (s.o.)

Satz 1 $\phi\phi$ -13 ϕ : Zeichnen eines schematischen Prismas.

Satz 135 : Warteschleife

Satz 14 ϕ -21 ϕ : Zeichnen schematischer Lichtstrahlen mit zeitlichen Verzögerungen durch eingebaute Warteschleifen

Satz 23 ϕ -245 : Zeichnen einer schwarzen Umrandung für das Spektrum (ebenfalls mit Warteschleifen)

Satz 25 ϕ -31 ϕ : Zeichnen des Spektrums, wiederum mit.

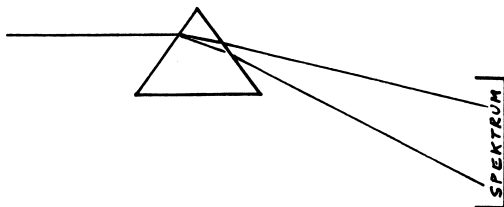
eingebauten Warteschleifen

Satz 32ø : Große Warteschleife, um dem Betrachter die Gelegenheit zu geben, die fertige Graphik in Ruhe zu studieren

Satz 33ø-34ø : Beendigung des Programms nach Löschen des Bildschirms und Hinweis auf die Möglichkeit des Neustarts

7. SCHRITT : Ergebnisse

Dieses Programm erzeugt ein Bild, welches schematisch der folgenden Skizze entspricht :



4.9

Das Ohm'sche Gesetz

①. SCHRITT : Vorstellung des Problems

In Stromleitern, in denen bei steigender Spannung die Stromstärke proportional zunimmt, gilt das Ohm'sche Gesetz.

Es besagt in Formelschreibweise :

$$\frac{\text{Spannung}}{\text{Stromstärke}} = \text{konstant} = R$$

Die Größe R nennt man den elektrischen Widerstand des betreffenden Stromleiters.

Im folgenden Programm soll für jeweils zwei eingegebene Werte (Spannung, Stromstärke, Widerstand) die jeweils dritte Größe berechnet werden.

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 9 : Ohm'sches Gesetz	Seite 174
-----------------------	--	--------------

2. SCHRITT : Problemanalyse

Das Programm, um das es hier geht, bietet unter mathematischen Gesichtspunkten überhaupt keine Schwierigkeiten, weil ja im Prinzip nur ein einziger Rechenschritt erforderlich ist.

Die einzige Schwierigkeit ist programmlogischer Art und besteht darin, daß die Eingabe aus drei unterschiedlichen Varianten bestehen kann, denen sich dann unterschiedliche Rechenprozeduren anschliessen.

Diese Eingaben können sein :

1. Spannung und Stromstärke;
2. Spannung und Widerstand;
3. Stromstärke und Widerstand.

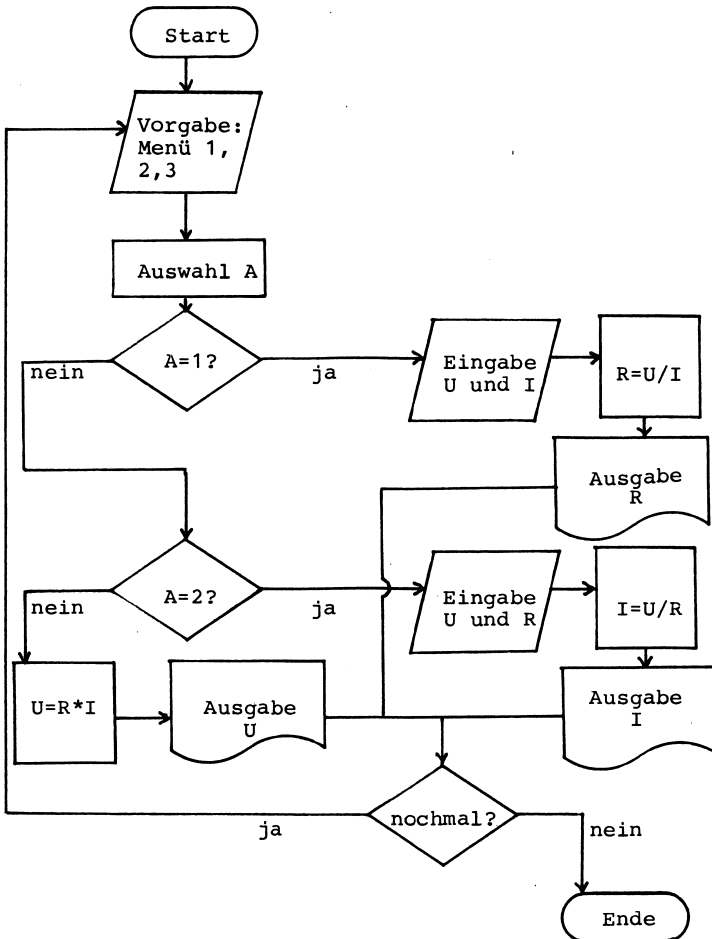
Dies gibt uns Gelegenheit, die sog. Menütechnik vorzuführen :

Es geht dabei darum, dem Programmbenutzer ein "Menü" zur Auswahl auf dem Bildschirm vorzugeben, aus dem dieser dann die gewünschte Variante auszuwählen hat. Je nach Auswahl kommen dann unterschiedliche, wenn auch hier sehr einfache Rechenprozeduren in Gang.

Dieses Menü würde hier aus drei Positionen bestehen (s.o).

Der Problemlösungsweg selbst wird aus dem folgenden Flußdiagramm deutlich.

3. SCHRITT : Flußdiagramm



VoB C64/ Schule	Kapitel 4 : Physik Abschnitt 9 : Ohm'sches Gesetz	Seite 176
-----------------------	--	--------------

4. SCHRITT : Programm

```

10 REM P6-OHMSCHES GESETZ
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR DARSTELLUNG DER BEZIEHUNGEN"
40 PRINT"ZWISCHEN STROMSTAERKE, SPANNUNG UND WI-"
50 PRINT:PRINTTAB(7)"STAND (OHMSCHES GESETZ). "
60 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984":PRINT
70 PRINT:PRINT"MIT WELCHEN VORGABEN WIRD GEARBEITET?":PRINT
80 PRINTTAB(5)"SPANNUNG UND STROMSTAERKE (1)"
90 PRINT:PRINTTAB(5)"SPANNUNG UND WIDERSTAND (2)"
100 PRINT:PRINTTAB(5)"STROMSTAERKE UND WIDERSTAND (3)"
110 PRINT:PRINT:INPUT"BITTE ZIFFER EINGEBEN " ;A
120 PRINTCHR$(147)
130 IF A>1 THEN 190
140 INPUT"SPANNUNG : ";U
150 INPUT"STROMSTAERKE : ";I
160 R=U/I:R=INT(R*100+.5)/100
170 PRINT:PRINT:PRINT"WIDERSTAND : ";R
180 GOTO 300
190 IF A=3 THEN 250
200 INPUT"SPANNUNG : ";U
210 INPUT"WIDERSTAND : ";R
220 I=U/R:I=INT(I*100+.5)/100
230 PRINT:PRINT:PRINT"STROMSTAERKE : ";I
240 GOTO 300
250 INPUT"WIDERSTAND : ";R
260 INPUT"STROMSTAERKE : ";I
270 U=R*I:U=INT(U*100+.5)/100
280 PRINT:PRINT:PRINT"SPANNUNG : ";U
300 PRINT:PRINT:PRINT:INPUT"NOCH EINE RECHNUNG ? (J/N) ";A$
310 IF A$="J" THEN PRINTCHR$(147):GOTO 70
320 PRINT:PRINT"ENDE DER BERECHNUNGEN":END

```

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 9 : Ohm'sches Gesetz	Seite 177
<p style="text-align: center;">5. SCHRITT : Variablenliste</p> <p>A = Auswahlvariable (Menü) A\$ = Stringvariable (Ja,Nein)</p> <p>I = Stromstärke R = Widerstand U = Spannung</p> <p style="text-align: center;">6. SCHRITT : Programmbeschreibung</p> <p>Satz 1ϕ-6ϕ : Überschrift und Erläuterungen</p> <p>Satz 7ϕ-1$\phi\phi$: Vorgabe des Menüs</p> <p>Satz 11ϕ : Menüauswahl</p> <p>Satz 12ϕ-13ϕ : Wenn A größer als 1 ist (A = Menü- variable) geht es nach Löschen des Bildschirms weiter bei Satz 19ϕ</p> <p>Satz 14ϕ-15ϕ : Eingaben gemäß Auswahl A = 1</p> <p>Satz 16ϕ-17ϕ : Berechnungen und Ergebnisausgabe</p> <p>Satz 18ϕ : Sprung zum Satz 3$\phi\phi$</p> <p>Satz 19ϕ-24ϕ : Entsprechende Berechnungen wie oben; nun aber für A = 2.</p>		

Voß C64/ Schule	Kapitel 4 : Physik Abschnitt 9 : Ohm'sches Gesetz	Seite 178
-----------------------	--	--------------

Satz 25 ϕ -28 ϕ : Entsprechende Berechnungen; nun für
A = 3

Satz 30 ϕ -31 ϕ : Abfrage, ob noch eine Berechnung ge-
wünscht wird

wenn ja, Löschen des Bildschirms und
zurück zum Satz 7 ϕ

Satz 32 ϕ : Beendigung des Programms

7. SCHRITT : Ergebnisse

Wählen wir bei der Menüauswahl z.B. die Ziffer 1, so
werden vom Programm Spannung und Stromstärke angefor-
dert. Geben wir daraufhin z.B. für die Spannung den
Wert 22 ϕ an und für die Stromstärke den Wert 11, so
erhalten wir als Ergebnisausdruck :

WIDERSTAND : 2 ϕ

NOCH EINE RECHNUNG (J/N) ?

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 1 : BASIC	Seite 179
-----------------------	---	--------------

5

Kapitel 5 : Sprachen
 =====

5.1 Vorbemerkung und BASIC-Ergänzungen

Auch beim Erlernen oder beim Üben von Fremdsprachen können Rechner wirkungsvolle Hilfe leisten. Dabei geht es jetzt nicht darum, komplette Sprachunterrichtsprogramme (zum Beispiel zum Erlernen von Englisch) vorzustellen, weil dies den Rahmen und den Anspruch dieses Buches weit übersteigen würde - außerdem kann man solche "Komplettprogramme" schon zu relativ günstigen Preisen käuflich erwerben. Vielmehr interessiert hier, mit geringem Aufwand einfache Programme zu erstellen, die den sprachlernenden Schüler bei seinen Bemühungen unterstützen können.

In Frage kommen hier in erster Linie solche Programme, die das Erlernen von Vokabeln erleichtern, oder die Übersetzungen bereitstellen, um so das Nachschlagen in Wörterbüchern zu ersparen.

Für derartige Programme ist es nun allerdings notwendig, einige zusätzliche BASIC-Anweisungen kennenzulernen :

Statement 16 :

```
nn GET Var.name$
```

Dieses Statement wartet, ähnlich wie das schon oft benutzte INPUT-Statement eine Informationseingabe ab, die hier aber nur aus der Betätigung einer einzigen Taste besteht und nicht durch die RETURN-Taste abgeschlossen werden muß. In den späteren Programmen wird sofort die Funktionsweise dieses Statements ersichtlich.

Häufig ist es sinnvoll, daß bestimmte Programmsegmente aus dem eigentlichen Hauptprogramm ausgelagert werden, um dann von diesem aus mehrfach angesprungen zu werden.

Solche ausgelagerten Teilprogramme nennt man Unterprogramme (subroutines). Der Sprung vom Hauptprogramm in das Unterprogramm erfordert das folgende Statement :

Statement 17 :

```
nn GOSUB mm
```

Dieses Statement bewirkt einen Sprung in das Unterprogramm, das mit der Satznummer mm beginnt.

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 1 : BASIC	Seite 181
<p>Ist dann das Unterprogramm abgearbeitet, muß ein Rücksprung in das Hauptprogramm erfolgen. Zu diesem Zweck wird das Unterprogramm mit dem folgenden Statement beschlossen :</p> <p><u>Statement 18</u> :</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> nn RETURN </div> <p>Dieses Statement bewirkt, daß ein Rücksprung an diejenige Stelle des Hauptprogramms erfolgt, die direkt hinter der entsprechenden GOSUB-Anweisung folgt.</p> <p>Auch die Wirkungsweise dieser beiden Statements wird in den folgenden Programmen deutlich werden, so daß hier auf Demonstrationsbeispiele verzichtet werden kann.</p>		

5.2 Englische unregelmäßige Verben

①. SCHRITT : Vorstellung des Problems

In einem ersten Beispiel soll ein Programm vorgestellt werden, welches für ein eingegebenes englisches Verb die dazugehörigen unregelmäßigen Stammformen angibt.

Beispielsweise soll der Rechner auf die Eingabe des Verbs

go

mit den Meldungen

go went gone (gehen)

antworten.

Um den Programmieraufwand nicht zu hoch werden zu lassen, wollen wir uns dabei auf 20 wichtige unregelmäßige Verben beschränken. Der Leser wird sofort erkennen, wie er das Programm in Eigenarbeit ergänzen muß, um einen kompletten Überblick über alle englischen unregelmäßigen Verben zu erhalten.

VoB C64/ Schule	Kapitel 5 : Sprachen Abschnitt 2 : Unregelmäßige Verben	Seite 183
-----------------------	--	--------------

② SCHRITT : Problemanalyse

Das Programm, um das es hier geht, muß für ein einzugebendes Schlüsselwort alle drei Stammformen und die deutsche Übersetzung des betreffenden Verbs ausgeben.

Dies macht es erforderlich, daß alle Informationen zunächst dem Rechner eingegeben werden.

Bei sehr umfangreichen Informationsbeständen ist es sinnvoll, die Ausgangsinformationen nicht bei jedem Programmstart neu zu lesen, sondern nur die gewünschte Information aus einer Textdatei einer entsprechend vorbereiteten Diskette interaktiv einzulesen. Diese Möglichkeit, auf die fortgeschrittenere Programmierer sicherlich zugreifen werden - zumindest sofern sie über ein Diskettenlaufwerk verfügen - soll hier aber nicht besprochen werden.

Wir lassen also bei jedem Programmstart die Ausgangsinformationen lesen und lassen dann das Programm das gewünschte englische Verb anfordern. Der Rechner vergleicht dann diese Eingabe mit den Gesamtinformationen, bis er das betreffende Verb im Datenbestand gefunden hat. Dann kann er die Stammformen und die deutsche Übersetzung ausgeben.

Findet er bei diesen Vergleichen in seinem Datenbestand das eingegebene Verb nicht, dann deshalb, weil es

- entweder im Datenbestand noch nicht vorhanden ist (wir haben ja vorläufig nur 20 Verben aufgenommen),

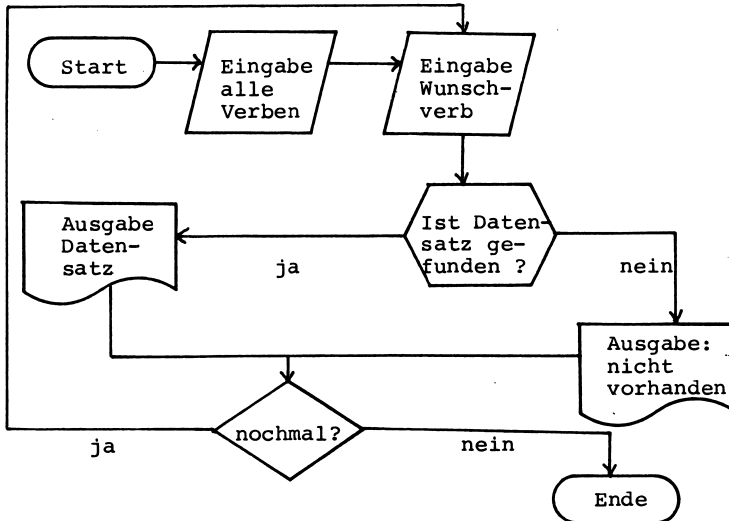
- oder, weil es sich bei dem eingegebenen Verb nicht um ein unregelmäßiges Verb handelt.

In beiden Fällen, die aber bei dem folgenden Programm wegen der Nicht-Vollständigkeit der Ausgangsinformationen nicht voneinander trennbar sind, ist eine Meldung auszugeben.

Weiterhin soll das Programm dem Benutzer die Gelegenheit geben, ein weiteres Verb abzufragen.

Die Stammformen selbst und die deutsche Übersetzung können, da es sich immer um Strings handelt, in einem doppelt indizierten String-Array gespeichert werden.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 2 : Unregelmäßige Verben	Seite 185
-----------------------	--	--------------

(4) SCHRITT : Programm

```

10 REM S1-UNREGELM.VERBEN,ENGLISCH
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR AUSGABE DER STAMMFORMEN"
40 PRINT"VON EINGEGEBENEN ENGLISCHEN UNREGEL-"
50 PRINTTAB(11)"MAESSIGEN VERBEN."
60 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"DIE ZAHL DER VORHANDENEN WORTE IST IN "
90 PRINT"DIESEM PROGRAMM BESCHRAENKT. ERWEITE-"
100 PRINT"RUNGEN SIND MOEGLICH, WENN DIE LISTE "
110 PRINT"DER DATA-STATEMENTS VERLAENGERT WIRD"
120 PRINT"(SATZ 500 FF.) UND WENN IN SATZ 160 DIE"
130 PRINT"ANGABE FUER N ERHOEHT WIRD.":PRINT:PRINT:PRINT
135 PRINT:PRINT
140 PRINTTAB(7)"BITTE EINE TASTE DRUECKEN"
150 GET A$:IF A$="" THEN 150
155 PRINTCHR$(147):PRINTTAB(12)"MOMENT, BITTE":PRINT:PRINT
160 N=20
170 DIM W$(N,4)
180 FOR I=1TON:FORJ=1TO4:READW$(I,J):NEXTJ:NEXTI
190 PRINTCHR$(147)
200 PRINT"BITTE, DAS DEUTSCHE VERB, DESSEN ENGLI-"
210 PRINT"SCHE STAMMFORMEN GEWUENSCHT WERDEN,"
220 INPUT"EINGEBEN : ";V$
230 FOR I=1 TO N
240 IF V$=W$(I,1) THEN 275
250 NEXT I
260 PRINT:PRINT:PRINT"VERB NICHT IM DATENBESTAND VORHANDEN,"
270 PRINT"BZW. NICHT UNREGELMAESSIG.":GOTO 310
275 PRINT:PRINT:PRINT
280 PRINT"VERB : ";W$(I,1):PRINT:PRINT:PRINT
290 PRINT"ENGLISCHE STAMMFORMEN : ":PRINT
300 FOR J=2 TO 4:PRINTW$(I,J);" ";:NEXTJ
310 PRINT:PRINT:PRINT:INPUT"NOCHMAL ?(J/N) ";A$
320 IF A$="J" THEN PRINTCHR$(147):GOTO 200
330 PRINT:PRINT"ENDE":END
500 DATA SEIN,BE,WAS,BEEN
502 DATA HABEN,HAVE,HAD,HAD
504 DATA BEGINNEN,BEGIN,BEGAN,BEGUN
506 DATA BRINGEN,BRING,BROUGHT,BROUGHT
508 DATA KOMMEN,COME,CAME,COME
510 DATA TUN/MACHEN,DO,DID,DONE
512 DATA ESSEN,EAT,ATE,EATEN
514 DATA KAUFEN,BUY,BOUGHT,BOUGHT
516 DATA FUEHLEN,FEEL,FELT,FELT

```

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 2 : Unregelmäßige Verben	Seite 186
-----------------------	--	--------------

518 DATA FINDEN,FIND,FOUND,FOUND
 520 DATA GEHEN,GO,WENT,GONE
 522 DATA VERLIEREN,LOSE,LOST,LOST
 524 DATA MACHEN,MAKE,MADE,MADE
 526 DATA LESEN,READ,READ,READ
 528 DATA LAUFEN,RUN,RAN,RUN
 530 DATA SAGEN,SAY,SAID,SAID
 532 DATA SEHEN,SEE,SAW,SEEN
 534 DATA SITZEN,SIT,SAT,SAT
 536 DATA SPRECHEN,SPEAK,SPOKE,SPOKEN
 538 DATA NEHMEN,TAKE,TOOK,TAKEN

5. SCHRITT : Variablenliste

A\$ = Stringvariable (Ja,Nein)

I = Laufvariable

J = Laufvariable

N = Anzahl der Datensätze

V\$ = Gewünschtes Verb

W\$ = Stammformen der Verben und deutsche Übersetzung
(Ausgangsdatenbestand)

VoB C64/ Schule	Kapitel 5 : Sprachen Abschnitt 2 : Unregelmäßige Verben	Seite 187
<p>6. SCHRITT : Programmbeschreibung</p> <p>Satz 1φ-135 : Überschrift, Erläuterungen und Hinweise, wie der Datenbestand aktualisiert werden kann</p> <p>Satz 14φ-15φ : Programm in "Warteposition". Die Programmausführung geht erst weiter, wenn der Benutzer irgendeine Taste drückt. Dies gelingt durch geeignete Verwendung des GET-Statements in Satz 15φ</p> <p>Satz 155 : Löschen des Bildschirms und Hinweis</p> <p>Satz 16φ-18φ : Dimensionierung und Einlesen der Daten</p> <p>Satz 19φ-22φ : Löschen des Bildschirms und Anforderung des gewünschten Verbs</p> <p>Satz 23φ-25φ : Suche nach dem richtigen Datenbestand Wird er gefunden, dann weiter bei 275 wird er nicht gefunden, weiter bei 26φ</p> <p>Satz 26φ-27φ : Meldung, daß Verb im Datenbestand nicht vorhanden; dann weiter bei Satz 31φ</p> <p>Satz 275-3φφ : Ausgabe der gesuchten Informationen</p> <p>Satz 31φ-32φ : Abfrage, ob weitere Ausgabe gewünscht wenn ja, Löschen des Bildschirms und zurück zu Satz 2φφ</p> <p>Satz 33φ : wenn nein, Ende des Programms</p>		

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 2 : Unregelmäßige Verben	Seite 183
-----------------------	--	--------------

Satz 500-538 : Daten

7. SCHRITT : Ergebnisse

Gibt man beispielweise auf die Anforderung des Rechners
aus Satz 200 ein :

GO

so antwortet der Rechner :

VERB : GEHEN (weiße Schrift)

ENGLISCHE STAMMFORMEN :

GO WENT GONE

NOCHMAL? (J/N) ?

Es ist deutlich zu sehen, wie das Programm geändert
werden müßte, wenn die Liste der unregelmäßigen Verben
komplettiert werden soll : In den Sätzen 500 ff. sind
weitere DATA-Statements anzuhängen und Satz 160 muß
ausgetauscht werden mit der Angabe der aktuellen Zahl
eingegebener Verben.

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 3 : Vokabeln	Seite 189
-----------------------	--	--------------

5.3 Französisch-Vokabeln

① SCHRITT : Vorstellung des Problems

In diesem Beispiel soll nun gezeigt werden, wie der Rechner als Vokabelheft-Ersatz Verwendung finden kann. Auf ein einzugebendes französisches Wort antwortet er mit dem entsprechenden deutschen Wort oder umgekehrt (es gilt hier die gleiche Anmerkung wie in Abschnitt 2.5 zur interaktiven Nutzung eines Diskettenspeichers).

Um das Programm wieder nicht zu groß werden zu lassen, beschränken wir uns auf nur zehn Vokabeln. Dieses Programm dient also nur als Beispiel. Wenn es echt genutzt werden soll, müßte sich der Leser die Mühe machen, seinen gesamten deutsch-französischen Vokabelvorrat zunächst dem Rechner einzugeben.

An den Funktionsprinzipien des Programms ändert sich aber dadurch nichts.

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 3 : Vokabeln	Seite 190
-----------------------	--	--------------

② SCHRITT : Problemanalyse

Bei diesem Problem können wir uns kurz fassen :

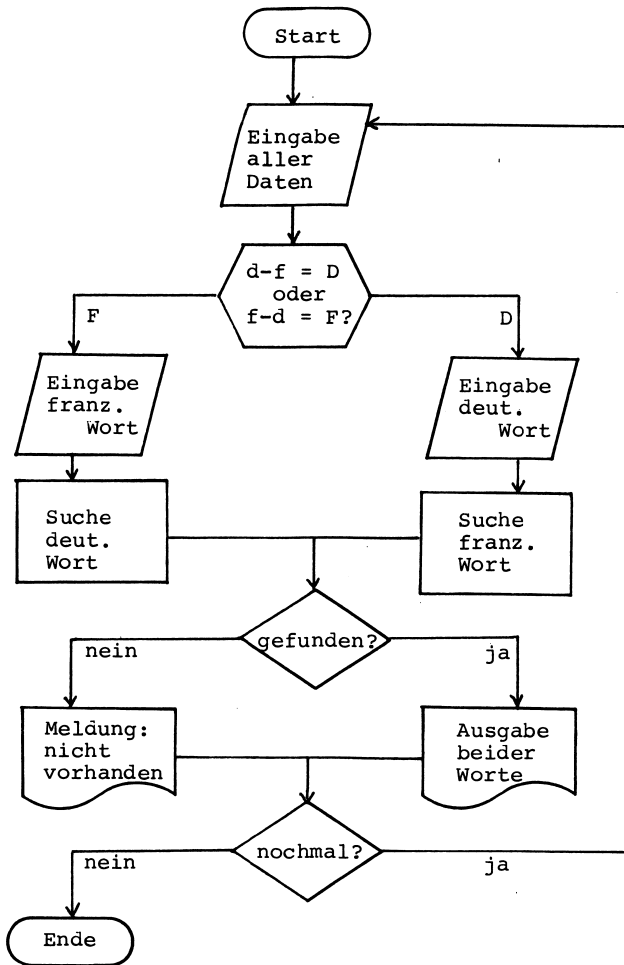
Gibt der Benutzer ein deutsches Wort ein, so muß der Rechner durch Vergleich mit allem ihm zur Verfügung stehenden deutschen (und damit paarweise verbundenen französischen Worten) das entsprechende deutsch-französische Wortpaar aus seinem Datenbestand herausuchen und ausgeben. Entsprechend ist auch im umgekehrten Fall zu verfahren.

Damit beide "Übersetzungsrichtungen" in nur einem Programm bewältigt werden können, muß dem Rechner zunächst mitgeteilt werden, ob deutsch-französische oder ob französisch-deutsche Übersetzung gewünscht wird.

Falls das eingegebene Wort nicht übersetzt werden kann, weil es in der (hier sehr kurzen) Vokabelliste (noch) nicht vorhanden ist, muß wieder eine entsprechende Meldung erfolgen.

Weiterhin sollte dem Benutzer Gelegenheit geboten werden, im gleichen Programmlauf eine weitere Übersetzung zu erhalten, wenn er dies wünscht.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 3 : Vokabeln	Seite 192
-----------------------	--	--------------

4. SCHRITT : Programm

```

10 REM S2-VERBEN DEUTSCH-FRANZOESISCH
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUM NACHSCHLAGEN VON VOKABELN:"
40 PRINT:PRINT"DEUTSCH/FRANZOESISCH ODER UMGEKEHRT."
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984":PRINT
60 PRINT:PRINT:PRINT:PRINT
70 PRINT"IM DATENBESTAND DIESES PROGRAMMS BEFIN-"
80 PRINT"DEN SICH NUR 10 VOKABELN.":PRINT
90 PRINT"SOLL DAS PROGRAMM AUSGEWEITET WERDEN, "
100 PRINT"SO MUESSEN IN 500 FF. WEITERE DATA-STA-"
110 PRINT"TEMENTS ANGEFUEGT UND DER WERT FUER N"
120 PRINT"IN STATEMENT 160 GEÄENDERT WERDEN."
130 PRINT:PRINT:PRINT:PRINTTAB(7)"BITTE EINE TASTE DRUECKEN."
140 GET A$:IF A$=""THEN 140
150 PRINTCHR$(147):PRINTTAB(7)"MOMENT BITTE "
160 N=10
170 DIM D$(N),F$(N)
180 FOR I=1TON:READD$(I),F$(I):NEXTI
190 PRINTCHR$(147)
200 PRINT"WELCHE UEBERSETZUNG WIRD GEWUENSCHT ?"
210 PRINT:PRINT:PRINT"-- DEUTSCH-FRANZOESISCH (1)"
220 PRINT:PRINT"ODER"
230 PRINT:PRINT "-- FRANZOESISCH-DEUTSCH (2)"
240 PRINT:PRINT:INPUT"BITTE 1 ODER 2 EINGEBEN ";Z
245 PRINTCHR$(147)
250 IF Z=2 THEN 350
260 PRINT:PRINT:INPUT"DEUTSCHES WORT BITTE : ";W$
280 FOR I=1 TO N:IF W$=D$(I) THEN 320
290 NEXT I
300 PRINT:PRINT"GESUCHTES WORT IM DATENBESTAND NICHT"
310 PRINT"VORHANDEN.":GOTO 450
320 PRINTCHR$(147):PRINT"DEUTSCH           : ";PRINT"☐";D$(I)
330 PRINT:PRINT"☐";:PRINT "FRANZOESISCH           : ";PRINT"☐";F$(I)
                                           :PRINT"☐"

340 GOTO 450
350 PRINT:PRINT:INPUT"FRANZ. WORT BITTE : ";W$
360 FORI=1TON:IFW$=F$(I)THEN 400
370 NEXT I
380 PRINT:PRINT"GESUCHTES WORT IM DATENBESTAND NICHT"
390 PRINT"VORHANDEN.":GOTO 450
400 PRINTCHR$(147):PRINT"FRANZOESISCH : ";PRINT"☐";F$(I)
410 PRINT:PRINT"☐"; "DEUTSCH           : ";PRINT"☐";D$(I):PRINT"☐"
450 PRINT:PRINT:PRINT:INPUT"NOCHMAL ? (J/N) ";A$
460 IF A$="J"THEN PRINTCHR$(147):GOTO 200
470 PRINT:PRINT"ENDE DER AUSGABE":END

```

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 3 : Vokabeln	Seite 193
<p style="text-align: center;"> 500 DATA KAUFEN,ACHETER 502 DATA HABEN,AVOIR 504 DATA GEHEN,ALLER 506 DATA SEHEN,VOIR 508 DATA TRINKEN,BOIRE 510 DATA MACHEN,FAIRE 512 DATA LESEN,LIRE 514 DATA ESSEN,MANGER 516 DATA NEHMEN,PRENDRE 518 DATA RAUCHEN,FUMER </p> <p style="text-align: center;"> (5) SCHRITT : Variablenliste </p> <p> A\$ = Stringvariable als Antwort bei GET bzw. Ja/Nein D\$ = deutsche Vokabeln F\$ = französische Vokabeln I = Laufindex N = Anzahl der Datensätze W\$ = gewünschtes Wort Z = Kennziffer bei der Menü-Auswahl </p>		

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 3 : Vokabeln	Seite 194
-----------------------	--	--------------

6. SCHRITT : Programmbeschreibung

- Satz 10-12 : Überschrift, Erläuterungen etc.
- Satz 13-15 : Warten, bis Tastendruck erfolgt und Löschen des Bildschirms
- Satz 16-18 : Angabe der Anzahl der Datensätze, Dimensionierung und Eingabe der Ausgangsdaten.
- Satz 19-24 : Löschen des Bildschirms und Menüvorgabe
- Satz 24-25 : Wird 2 eingegeben (französisch-deutsch), dann weiter bei Satz 35
- Satz 26 : Eingabe eines deutschen Wortes
- Satz 28-29 : Suche nach dem entsprechenden Datensatz
- Satz 30-31 : Ist das gesuchte Wort im Datenbestand (noch) nicht vorhanden, weiter bei 45
- Satz 32-33 : Ausgabe der Ergebnisse
- Satz 34 : Sprung nach Satz 45
- Satz 35-41 : Entsprechende Vorgehensweise für die Eingabe eines französischen Wortes
- Satz 45-47 : Beendigung des Programms mit Wiederholungsmöglichkeit
- Satz 50-51 : Daten

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 3 : Vokabeln	Seite 195
-----------------------	--	--------------

7. SCHRITT : Ergebnisse

Geben wir nach der Menüvorgabe die Ziffer 1 ein und entscheiden uns damit für die deutsch-französische Ausgabe, so fordert der Rechner von uns ein deutsches Wort an.

Geben wir z.B. daraufhin das Wort haben ein, so antwortet der Rechner :

DEUTSCH : HABEN (in weißer Schrift)

FRANZÖSISCH : AVOIR (ebenfalls in weißer Schrift)

NOCHMAL ? (J/N) ?

5.4 Englisch-Vokabeltest

1. SCHRITT : Vorstellung des Problems

Ähnlich wie im vorangegangenen Abschnitt sollen - ausgehend von einem bestimmten Vokabelvorrat - Übersetzungen bereitgestellt werden. Im Gegensatz aber zum vorhergehenden Programm soll der Benutzer aufgefordert werden, zu einem vom Rechner zufällig ausgewählten deutschen Wort das korrekte englische Wort einzusetzen.

Es wird hier also ein Programm vorgestellt, das so funktioniert wie früher das Vokabellernen mit zugehaltener Fremdwortspalte im Vokabelheft vor sich ging.

Es leuchtet unmittelbar ein, daß ein entsprechendes Programm auch für Englisch-Deutsch bzw. für andere Fremdsprachen verwendet werden könnte.

Im Interesse der Reduzierung des Programmieraufwandes beschränken wir uns hier wieder auf die exemplarische Darstellung anhand von nur zehn Vokabeln. Eine Untergruppe davon wird jeweils vom Rechner zufällig ausgewählt und dem Benutzer wird nach jeder Serie mitgeteilt, wie hoch seine Erfolgsquote war.

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 4 : Vokabeltest	Seite 197
-----------------------	---	--------------

②. SCHRITT : Problemanalyse

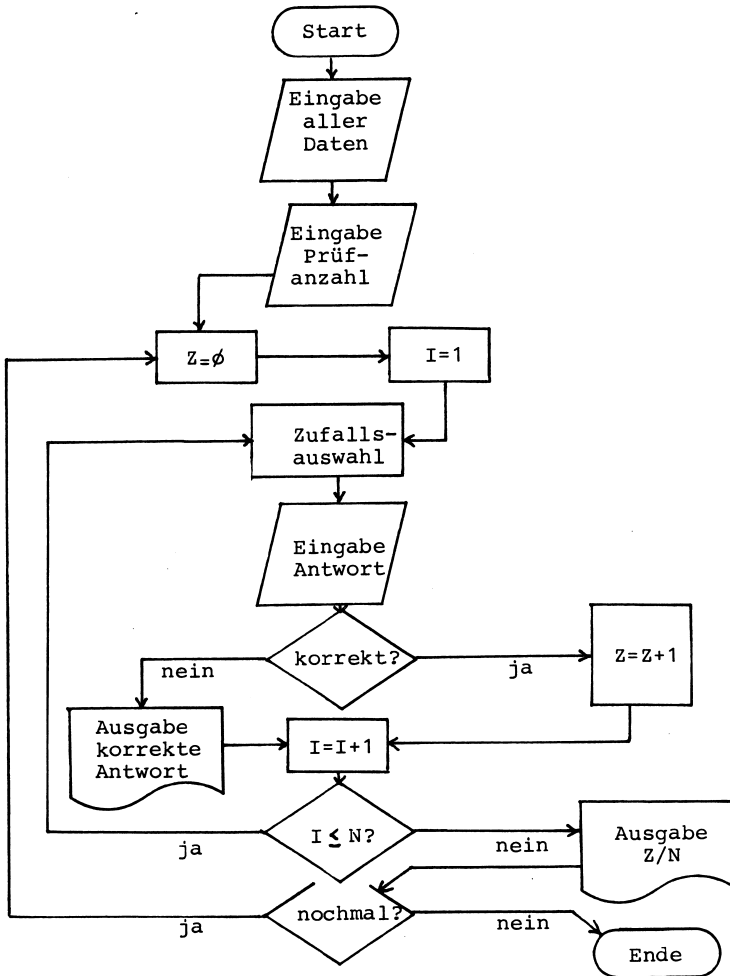
Ausgehend von einem vorgegebenen Datenbestand, muß das Programm eine Zufallsauswahl treffen.

Auf jede vorgegebene Vokabel antwortet der Benutzer. Die Anzahl der richtigen Antworten wird in Beziehung gesetzt zur Anzahl der angeforderten Antworten.

Das Programm soll Gelegenheiten zur Testwiederholung bieten.

Besondere Schwierigkeiten tauchen bei der Problemanalyse desweiteren nicht auf.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 4 : Vokabeltest	Seite 199
-----------------------	---	--------------

④ SCHRITT : Programm

```

10 REM S3-VOKABELTEST
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUM ABFRAGEN VON VOKABELN."
40 PRINT:PRINT"HIER : DEUTSCH/ENGLISCH ODER UMGEGEHERT"
50 PRINT:PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT"IM DATENBESTAND DIESES PROGRAMMS SIND "
80 PRINT"NUR 10 VOKABELN.":PRINT
90 PRINT"SOLL DAS PROGRAMM ERWEITERT WERDEN, SO"
100 PRINT"MUESSEN IN 1000 FF. WEITERE DATA-STATE-"
110 PRINT"MENTS ANGEFUEGT UND DIE ZAHL N IN SATZ"
120 PRINT"NR. 160 ERHOEHT WERDEN."
130 PRINT:PRINT:PRINTTAB(5)"BITTE EINE TASTE DRUECKEN"
140 GET A$:IF A$="" THEN 140
145 PRINTCHR$(147)
150 PRINTTAB(7)"MOMENT BITTE"
160 N=10
170 DIM D$(N,2)
180 FORI=1TON:READ D$(I,1),D$(I,2):NEXTI
185 ZZ=1
190 PRINTCHR$(147)
200 PRINT"DEUTSCH/ENGLISCH (1)"
210 PRINT:PRINT:PRINT"ODER":PRINT:PRINT
220 PRINT"ENGLISCH/DEUTSCH (2)"
230 PRINT:PRINT:PRINT"BITTE 1 ODER 2 EINGEBEN":PRINT
240 INPUT"           ";Z
245 IF Z<>ZZ THEN GOSUB 3000:REM TAUSCH
250 PRINT:PRINT:PRINT"WIEVIELE VOKABELN SOLLEN GEPRUEFT WER-"
260 INPUT"DEN ";A
270 GOSUB 2000
400 PRINT:INPUT"NOCHMAL ? (J/N) ";A$
410 IF A$="J" THEN PRINTCHR$(147):ZZ=Z:GOTO 200
420 PRINT:PRINT"ENDE DER AUSGABE":END
1000 DATA LAUFEN,RUN,SCHREIBEN,WRITE,DRUCKEN,PRINT,GEHEN,GO
1010 DATA WENN,IF,UNTERPROGRAMM,SUBROUTINE,RECHNER,COMPUTER
1020 DATA ZEICHEN,CHARACTER,BILDSCHIRM,SCREEN,TASTE,KEY
2000 PRINTCHR$(147)
2005 K=0
2010 I=1
2020 R=INT(RND(1)*N+1)
2030 PRINTD$(R,1):INPUT"          ANTWORT      : ";W$
2040 IF W$=D$(R,2) THEN K=K+1:PRINT:PRINT:PRINTTAB(10)"SEHR GUT !"
:GOTO 2070

```

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 4 : Vokabeltest	Seite 200
-----------------------	---	--------------

```

2050 PRINT:PRINT:PRINT"DAS WAR LEIDER FALSCH."
2060 PRINT:PRINT"RICHTIG MUSS ES HEISSEN : ";D*(R,2)
2070 PRINT:PRINT:PRINT:I=I+1
2080 IF I<=A THEN 2020
2090 KA=(K/A)*100
2100 PRINT:PRINT"ANTEIL KORREKTER ANTWORTEN : ";KA;" %"
2110 RETURN
3000 REM UP TAUSSCH
3010 FOR I=1TON
3020 H#=D*( I,1):D*( I,1)=D*( I,2):D*( I,2)=H#
3030 NEXT I
3040 RETURN

```

(5) SCHRITT : Variablenliste

- A = Anzahl der zu prüfenden Vokabeln
- A\$ = Antwortstring (Ja, Nein), bzw. im GET-Statement

- D\$ = Vokabeln
- H\$ = Hilfsfeld beim Tausch in UP 3000

- I = Laufindex
- K = Anzahl korrekter Antworten
- KA = Anteil korrekter Antworten
- N = Anzahl der Vokabeln im Datenbestand
- R = Zufallszahl

- W\$ = Antwort des Benutzers

- Z = Zifferantwort im Menü
- ZZ = Hilfsgröße (auf 1 gesetzt) für den eventuellen Tausch Deutsch-Englisch

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 4 : Vokabeltest	Seite 201
-----------------------	---	--------------

⑥ SCHRITT : Programmbeschreibung

Satz 10-12 : Überschrift, Erläuterungen und Hinweise auf die eventuelle Verlängerung des Programms

Satz 13-15 : Warten, Löschen des Bildschirms und Hinweis auf die Einlesedauer

Satz 16-18 : Angabe der Zahl der Vokabeln im Datenbestand, Dimensionieren und Einlesen.

Satz 185 : Belegung der Hilfsvariablen ZZ mit 1

Satz 19-24 : Löschen des Bildschirms, Ausgabe des Menüs und Anforderung der Benutzerauswahl

Satz 245 : Abfrage, ob Menü 1 gewählt wurde
wenn nein, Sprung ins UP 3000
wenn ja, weiter bei 250

Satz 25-26 : Anforderung der Anzahl der zu testenden Vokabeln

Satz 27 : Sprung ins UP 2000

Satz 40-42 : Abfrage, ob noch ein Test (zurück zu Satz 20) und Beendigung des Programms

Satz 100-102: Daten

VoB C64/ Schule	Kapitel 5 : Sprachen Abschnitt 4 : Vokabeltest	Seite 202
-----------------------	---	--------------

Satz 2000-2110 : UP Zufallsauswahl :

- 2005-2010 : Zähler K und I auf Anfangspositionen
- 2020 : Zufallszahl
- 2030 : Ausgabe der Vokabel und Anforderung der Antwort
- 2040 : Bei korrekter Antwort erfolgt Sprung nach 2070
- 2050-2060 : Fehlermeldung und Ausgabe der korrekten Antwort
- 2070-2080 : Nächste Vokabel
- 2090-2100 : Ausgabe des Erfolgsanteils
- 2110 : Rücksprung

Satz 3000-3040 : UP Tausch :

Tausch Deutsch-Englisch - Englisch-Deutsch

⑦. SCHRITT : Ergebnisse

Ergebnisse brauchen hier nicht vorgestellt zu werden. Was im Laufe des Programmablaufs an Ergebnissen ausgegeben wird, ergibt sich unmittelbar aus der Programmbeschreibung (siehe insbesondere Satz 2040 bis Satz 2100).

5.5 Das Sortieren von Vokabeln

1. SCHRITT : Vorstellung des Problems

Das Sortieren von Datenbeständen ist eines der wichtigsten Einsatzgebiete von Rechnern. Das gilt sowohl für das Sortieren von Zahlen als auch für das alphabetische Sortieren von Begriffen oder allgemein von Strings.

Es gibt eine ganze Reihe unterschiedlicher Sortieralgorithmen, die sich vor allem letztlich in der Sortiergeschwindigkeit voneinander unterscheiden.

Dieser Aspekt der Rechengeschwindigkeit soll hier nicht interessieren - vielmehr wollen wir einen Algorithmus vorstellen, der besonders klar den wesentlichen Grundgedanken, dem jeder Sortierprozeß folgen muß, zum Ausdruck bringt.

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 5 : Sortieren	Seite 204
-----------------------	---	--------------

② SCHRITT : Problemanalyse

"Sortieren" bedeutet, daß wir je zwei Strings aus einem Datenbestand miteinander vergleichen :

Wir vergleichen den ersten Schritt für Schritt mit allen übrigen Strings. Folgt dabei der erste String im Alphabet hinter dem zweiten, dann müssen die beiden gerade verglichenen Strings getauscht werden.

Dadurch wird erreicht, daß dann, wenn der erste String mit allen übrigen verglichen wurde (nach der ersten Runde also), an der ersten Stelle nun derjenige String steht, der im Alphabet als erster kommt.

Dann vergleichen wir den jetzt an zweiter Stelle stehenden String mit allen übrigen (außer mit dem ersten String). Auch hier wird wieder getauscht, wenn es erforderlich ist. Am Schluß der zweiten Vergleichsrunde steht dann an der zweiten Stelle derjenige String, der im Alphabet tatsächlich an zweiter Stelle kommt.

Dann vergleichen wir den jetzt an dritter Stelle stehenden String mit allen übrigen ... usw.

Es werden insgesamt so viele Runden durchlaufen, wie überhaupt Vergleiche möglich sind. Haben wir beispielsweise fünf Strings zu sortieren, so müssen vier derartige Runden durchlaufen werden : In der ersten Runde gibt es vier Vergleiche, in der zweiten Runde drei, in der dritten Runde zwei und in der vierten Runde noch einen Vergleich.

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 5 : Sortieren	Seite 205
-----------------------	---	--------------

Schematisch sieht dies folgendermaßen aus :

- 1. Runde : Vergleich Feld 1 mit Feld 2
Vergleich Feld 1 mit Feld 3
Vergleich Feld 1 mit Feld 4
Vergleich Feld 1 mit Feld 5

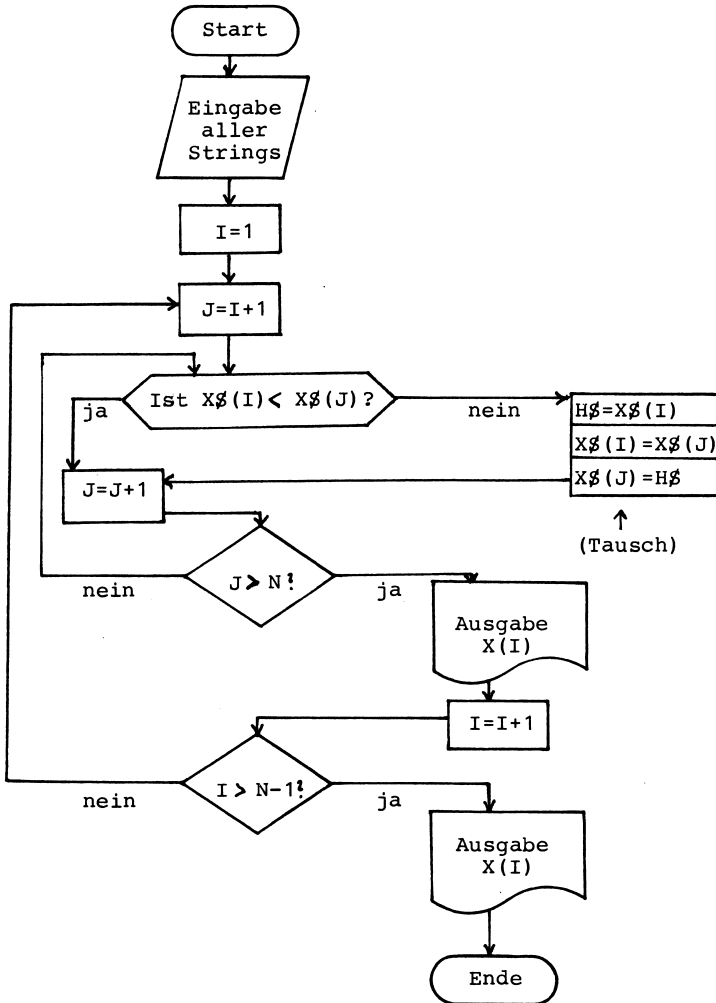
- 2. Runde : Vergleich Feld 2 mit Feld 3
Vergleich Feld 2 mit Feld 4
Vergleich Feld 2 mit Feld 5

- 3. Runde : Vergleich Feld 3 mit Feld 4
Vergleich Feld 3 mit Feld 5

- 4. Runde : Vergleich Feld 4 mit Feld 5

Es ist dabei zu beachten, daß nach den ersten Vergleichen die Inhalte der Felder immer andere sein können, je nachdem, ob und an welchen Stellen getauscht werden mußte.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 5 : Sortieren	Seite 207
-----------------------	---	--------------

4. SCHRITT : Programm

Im Gegensatz zu dem im vorangegangenen Flußdiagramm vorgesehenen Ablauf ist hier im Programm zusätzlich vorgesehen, daß der Benutzer entscheiden kann, ob er die deutsch-englischen Vokabeln alphabetisch nach der deutschen Schreibweise oder nach der englischen Schreibweise sortieren möchte :

```

10 REM S4-SORT
20 PRINTCHR$(147)
30 PRINTTAB(2)"PROGRAMM ZUM SORTIEREN VON VOKABELN."
40 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
50 PRINT:PRINT:PRINT:PRINT
60 PRINT"IN DIESEM PROGRAMM WERDEN NUR 10 ENGLI-"
70 PRINT"SCHE VOKABELN SORTIERT.":PRINT
80 PRINT"SOLLEN ANDERE VOKABELN SORTIERT WERDEN,"
90 PRINT"SO MUESSEN DIE DATA IN 500 FF. UND N IN"
100 PRINT"SATZ 150 GEÄNDERT WERDEN."
110 PRINT:PRINT:PRINT:PRINT:PRINT
120 PRINT"BITTE IRGEND EINE TASTE DRUECKEN !"
130 GETA$:IF A$="" THEN 130
140 PRINTCHR$(147)
150 N=10
160 DIM E$(N),D$(N)
170 FOR I=1 TO N:READ E$(I),D$(I):NEXT I
180 PRINT"SOLL NACH DEN ENGLISCHEN ODER NACH DEN"
190 PRINT"DEUTSCHEN VOKABELN SORTIERT WERDEN ?"
200 PRINT:INPUT"GIB E ODER D EIN : ";A$
210 IF A$="D" THEN 240
220 PRINTCHR$(147):PRINT"ENGLISCH";TAB(20)"DEUTSCH":PRINT:PRINT
230 GOSUB 1000:REM UP SORT
235 GOTO 270
240 FOR I=1 TO N:H$=E$(I):E$(I)=D$(I):D$(I)=H$:NEXT I
250 PRINTCHR$(147):PRINT"DEUTSCH";TAB(20)"ENGLISCH":PRINT:PRINT
260 GOSUB 1000:REM UP SORT
270 PRINT:PRINT"ENDE DER AUSGABE":END

```

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 5 : Sortieren	Seite 208
-----------------------	---	--------------

```

500 DATA PRINT,DRUCKEN,WRITE,SCHREIBEN,END,ENDE,GO,GEHEN
510 DATA RUN,LAUFEN,DATA,DATEN,SUBROUTINE,UNTERPROGRAMM
520 DATA GET,BEKOMMEN,SAVE,SPEICHERN,LOAD,LADEN
1000 REM UP SORT
1010 FOR I=1 TO N-1
1020 FOR J=I+1 TO N
1030 IF E$(I)<=E$(J) THEN 1070
1040 H$=E$(I):G$=D$(I)
1050 E$(I)=E$(J):D$(I)=D$(J)
1060 E$(J)=H$:D$(J)=G$
1070 NEXT J
1080 PRINT$(I);TAB(20)D$(I)
1090 NEXT I
1100 PRINT$(I);TAB(20)D$(I)
1110 RETURN

```

5. SCHRITT : Variablenliste

A\$ = Stringvariable für Antworten

D\$ = Deutsche Vokabeln

E\$ = Englische Vokabeln

G\$ = Hilfsfeld1

H\$ = Hilfsfeld2

I = Laufvariable

J = Laufvariable

N = Anzahl der Vokabeln

Voß C64/ Schule	Kapitel 5 : Sprachen Abschnitt 5 : Sortieren	Seite 209
-----------------------	---	--------------

⑥ SCHRITT : Programmbeschreibung

- Satz 1φ-11φ : Überschrift und Erläuterungen
- Satz 12φ-14φ : Warten und Löschen des Bildschirms
- Satz 15φ-17φ : Angabe der Zahl der Vokabeln, Dimensionierung und Einlesen der Vokabeln
- Satz 18φ-21φ : Abfrage, ob nach deutschen oder nach englischen Worten sortiert werden soll

falls nach deutschen Worten, weiter bei 27φ; andernfalls weiter bei 22φ
- Satz 22φ : Ausgabe einer Tabellenüberschrift
- Satz 23φ : Sprung ins UP 1φφφ zum Sortieren
- Satz 235 : Sprung zum Satz 27φ
- Satz 24φ : Austausch der deutschen und der englischen Vokabeln
- Satz 25φ : Ausgabe einer Tabellenüberschrift
- Satz 26φ : Sprung ins UP 1φφφ zum Sortieren
- Satz 27φ : Beendigung des Programms
- Satz 5φφ-52φ : Daten

VoB C64/ Schule	Kapitel 5 : Sprachen Abschnitt 5 : Sortieren	Seite 210
-----------------------	---	--------------

Satz 1000-1110 : UP Sortieren :

Unterprogramm zum Sortieren von
Strings :

1010 : Rundenzähler
1020 : Vergleichszähler
1030 : Vergleich
fällt er befriedigend aus,
weiter bei 1070, sonst
weiter bei 1040
1040-1060 : Tausch
1070 : Nächster Vergleich
1080 : Nachdem kein Vergleich
mehr möglich ist, Ausgabe
der beiden nun an erster
Stelle stehenden Strings
1090 : Nächste Runde
1100 : Ausgabe des letzten String-
paares
1110 : Rücksprung

7. SCHRITT : Ergebnisse

Wünschen wir beispielsweise die Sortierung nach den
deutschen Vokabeln, so erhalten wir :

DEUTSCH	ENGLISCH
BEKOMMEN	GET
DATEN	DATA
DRUCKEN	PRINT
ENDE	END
GEHEN	GO
...	...

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 1 : Vorbemerkung	Seite 211
-----------------------	---	--------------



Kapitel 6 : Biologie/Ökologie
=====

6.1 Vorbemerkung

Der Rechnereinsatz bei biologisch-ökologischen Problemen bezieht sich insbesondere darauf, Entwicklungs- und Wachstumsvorgänge zu simulieren (natürlich könnten aber auch Abfrageprogramme nach dem Muster des vorangegangenen Kapitels entwickelt werden). Diese Entwicklungen können dann auf dem Bildschirm tabellarisch oder auch graphisch ausgegeben werden.

6.2 Ungebremstes Wachstum

①. SCHRITT : Vorstellung des Problems

Eine Population mit dem Anfangsbestand X_0 wachse mit einer jährlichen Wachstumsrate von R Prozent (R sei positiv). Wenn dieses Wachstum nicht an äußere Grenzen stößt - was aber auf lange Sicht natürlich sehr unwahrscheinlich ist - dann stellt sich ein sog. exponentielles Wachstum ein.

Der Ablauf eines solchen sehr einfachen Wachstumsprozesses soll durch ein BASIC-Programm simuliert werden.

②. SCHRITT : Problemanalyse

Für das entsprechende Programm müssen der Anfangsbestand und die jährliche Zuwachsrate eingegeben werden.

Nach einem Jahr erhält man dann den Bestand X_1 :

$$X_1 = X_0 + X_0 * R / 100 = X_0 * (1 + R / 100)$$

Nach zwei Jahren ergibt sich X_2 :

$$X_2 = X_1 * (1 + R / 100) = X_0 * (1 + R / 100) * (1 + R / 100)$$

Daraus folgt :

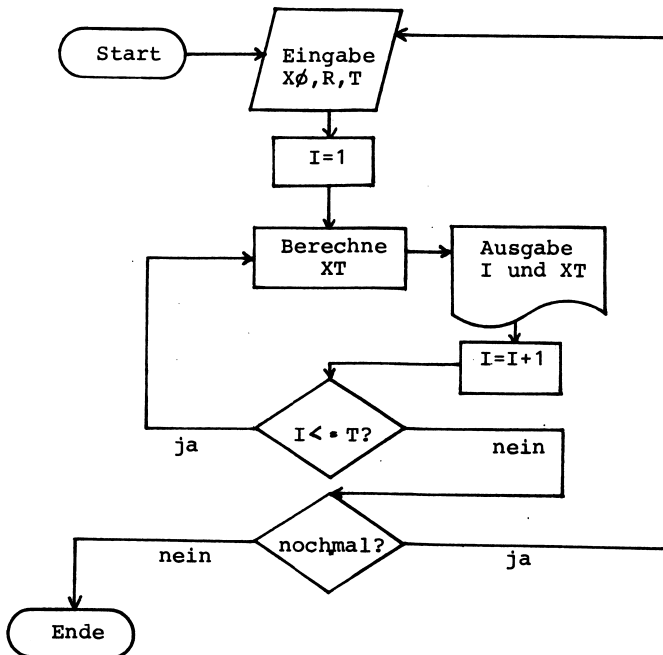
$$X_2 = X_0 * (1 + R/100)^2$$

Generell gilt deshalb nach T Jahren :

$$X_T = X_0 * (1 + R/100)^T$$

Diese Formel zeigt den Rechenalgorithmus für das entsprechende BASIC-Programm.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 2 : Wachstum	Seite 214
-----------------------	---	--------------

4. SCHRITT : Programm

```

10 REM B1-WACHSTUM
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR TABELLARISCHEN DARSTELLUNG"
40 PRINTTAB(8)"UNGEBREMSTEN WACHSTUMS"
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT
70 INPUT"ANFANGSBESTAND      : ";X0
80 PRINT:INPUT"WACHSTUMSRATE IN % : ";R
90 PRINT:INPUT"ANZAHL DER JAHRE : ";T
100 PRINTCHR$(147):PRINT"NACH...JAHREN  BESTAND":PRINT:PRINT
110 I=1
120 XT=X0*(1 + R/100)^I:XT=INT(XT*100+.5)/100
130 PRINTTAB(5)I;TAB(17)XT
140 I=I+1
145 IF I/10=INT(I/10) THEN PRINT:PRINT"BITTE CONT EINGEBEN":STOP
150 IF I<=T THEN 120
160 PRINT:PRINT:INPUT"NOCHMAL ? (J/N) ";A$
170 IF A$="J" THEN PRINTCHR$(147):GOTO 70
180 PRINT:PRINT:PRINT"ENDE DER AUSGABE":END

```

5. SCHRITT : Variablenliste

A\$ = Antwortstring (Ja,Nein)

I = Laufindex

R = Wachstumsrate in %

T = Zeit

X0 = Anfangsbestand

XT = Bestand zum Zeitpunkt T

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 2 : Wachstum	Seite 215
<p>6. SCHRITT : Programmbeschreibung</p> <p>Satz 1ϕ-6ϕ : Überschrift</p> <p>Satz 7ϕ-9ϕ : Vorgabe der Input-Informationen, die das Programm benötigt</p> <p>Satz 1$\phi\phi$: Ausgabe einer Tabellenüberschrift</p> <p>Satz 11ϕ : Vorgabe der ersten Periode</p> <p>Satz 12ϕ : Berechnung des jeweiligen Bestandes</p> <p>Satz 13ϕ : Ausgabe</p> <p>Satz 14ϕ : Übergang zur nächsten Periode</p> <p>Satz 145 : Nach jeder zehnten Periode wird das Programm unterbrochen (also dann, wenn I durch 1ϕ ohne Rest teilbar ist); Fortsetzung ist nur mit dem Kommando CONT möglich, deshalb ein entsprechender Hinweis</p> <p>Satz 15ϕ : Solange I kleiner gleich T ist, ist das Programm fortzusetzen (Satz 12ϕ)</p> <p>Satz 16ϕ-18ϕ : Beendigung des Programms, es sei denn es wird ein zusätzlicher Programmlauf gewünscht (dann nach Löschen des Bildschirms zurück zu Satz 7ϕ)</p>		

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 2 : Wachstum	Seite 216
-----------------------	---	--------------

7. SCHRITT : Ergebnisse

Geben wir z.B. als Anfangsbestand den Wert 500 ein,
als Wachstumsrate den Wert 4 (%) und als Zeitdauer
den Wert 8 (z.B. Jahre), so erzeugt das Programm die
folgende Ausgabe :

NACH ... JAHREN	BESTAND
1	520
2	540.8
3	562.43
4	584.93
5	608.33
6	632.66
7	657.97
8	684.28

NOCHMAL ? (J/N) ?

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 3 : Gebremstes Wachstum	Seite 217
-----------------------	--	--------------

6.3 Gebremstes Wachstum

①. SCHRITT : Vorstellung des Problems

Exponentielles Wachstum, wie es im vorangegangenen Abschnitt vorgestellt wurde, gibt es in der Realität praktisch nicht oder nur in sehr beschränkten Zeitabschnitten. Es muß vielmehr mit hemmenden Effekten gerechnet werden, die in der Regel um so stärker werden, je größer die Werte der interessierenden Variablen schon geworden sind.

Man denke beispielsweise daran, daß eine exponentiell wachsende Bevölkerung sehr rasch an Ernährungsgrenzen stößt, die dem weiteren Wachstum immer größere Hemmnisse auferlegen.

Eine solcherart "gedämpfte" Entwicklung soll im folgenden BASIC-Programm simuliert werden.

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 3 : Gebremstes Wachstum	Seite 218
-----------------------	--	--------------

(2.) SCHRITT : Problemanalyse

Die Problemanalyse führt hier zu einem ähnlichen Lösungsweg wie im vorangegangenen Abschnitt :

Zunächst gilt auch hier, daß Ausgangswerte benötigt werden :

- X_0 = Ausgangsbestand
- R_0 = Jährliche Zunahmerate (in %) zu Beginn des Entwicklungsprozesses

Im Gegensatz zum Beispiel zuvor gehen wir jetzt aber davon aus, daß die jährliche Zunahmerate nicht konstant bleibt, sondern um so kleiner wird, je größer die Population schon geworden ist.

Da die Population im Zeitablauf wächst, können wir in einem sehr einfachen Denkmodell also einfach davon ausgehen, daß die Zuwachsraten mit fortschreitender Zeit ab sinkt. Beispielsweise können wir die Zuwachsraten bei Verdopplung der Zeit halbieren, bei Vervierfachung der Zeit auf ein Viertel zurückgehen lassen usw.

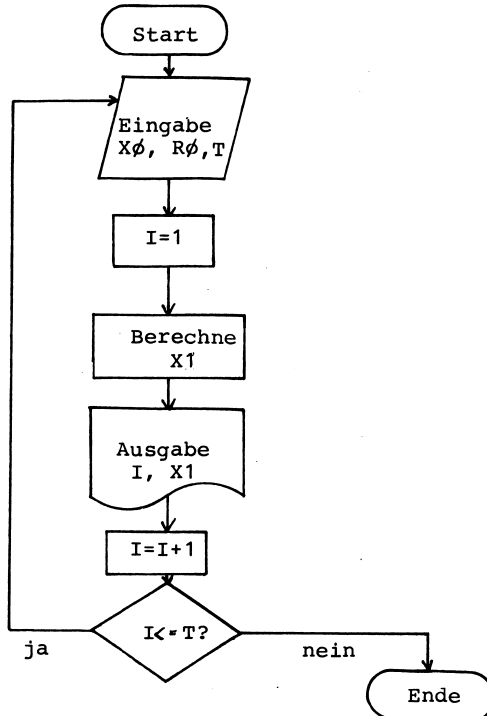
Natürlich lassen sich auch andere mathematische Modelle vorstellen, um diesen Bremsvorgang zu simulieren - darauf kommt es aber ja in diesem Zusammenhang nicht entscheidend an.

Den so beschriebenen Rechenalgorithmus können wir mathematisch folgendermaßen fassen :

$$X_1 = X_0 * (1 + R/I)$$

Das heißt, der Bestand der folgenden Periode X_1 berechnet sich aus dem der vorhergehenden Periode X_0 so wie im letzten Beispiel; die Wachstumsrate aber wird durch den Laufindex I dividiert, so daß sie um so kleiner wird, je höher I wird, d.h. je weiter die Zeit schon fortgeschritten ist.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 3 : Gebremstes Wachstum	Seite 220
-----------------------	--	--------------

④ SCHRITT : Programm

```

10 REM B2-GEbremSTES WACHSTUM
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR TABELLARISCHEN DARSTELLUNG"
40 PRINTTAB(2)"EINES GEBREMSTEN WACHSTUMSPROZESSES."
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT:PRINT
70 INPUT"ANFANGSBESTAND      : ";X0
80 PRINT:INPUT"W.RATE IM 1.JAHR % : ";R
85 PRINT:INPUT"WIEVIELE JAHRE    : ";T
90 PRINTCHR$(147)
100 PRINT"NACH...JAHREN  BESTAND  ZUWACHS (%)":PRINT:PRINT
110 I=1
120 X1=X0+X0*(R/I)/100
130 X1=INT(X1*100+.5)/100
140 Z=(X1-X0)/X0:Z=Z*100:Z=INT(Z*100+.5)/100
150 PRINTTAB(5)I;TAB(16)X1;TAB(28)Z
160 X0=X1:I=I+1:IF I<=T THEN 120
180 PRINT:INPUT"NOCHMAL ? (J/N) ";A$
190 IF A$="J" THEN PRINTCHR$(147):GOTO 70
200 PRINT:PRINT"ENDE DER AUSGABE":END

```

⑤ SCHRITT : Variablenliste

A\$ = Antwortstring (Ja, Nein)
I = Laufindex (Anzahl der Perioden)
R = Wachstumsrate zu Beginn des Prozesses
T = Zeitdauer des Prozesses

X0 = Anfangsbestand.
X1 = Bestand nach einer Periode
Z = Bestandsveränderung

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 3 : Gebremstes Wachstum	Seite 221
<p style="text-align: center;">6. SCHRITT : Programmbeschreibung</p> <p>Satz 1ϕ-6ϕ : Überschrift und Erläuterungen</p> <p>Satz 7ϕ-9ϕ : Anforderung der Ausgangsdaten und Löschen des Bildschirms</p> <p>Satz 1$\phi\phi$: Ausgabe einer Tabellenüberschrift</p> <p>Satz 11ϕ : Vorgabe des Periodenzählers</p> <p>Satz 12ϕ-15ϕ : Berechnung des Periodenbestands, des Zuwachses und Ausgabe der entsprechenden Werte nach Rundung auf zwei Dezimalstellen</p> <p>Satz 16ϕ : Belegung von Xϕ mit X1 und Fortführung der Berechnungen, solange I nach Erhöhung um 1 kleiner als der vorgegebene Zeitwert T bleibt</p> <p>Satz 18ϕ-2$\phi\phi$: Abfrage, ob noch eine Berechnung gewünscht (wenn ja, zurück nach 7ϕ) und Beendigung des Programms</p>		

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 3 : Gebremstes Wachstum	Seite 222
-----------------------	--	--------------

7. SCHRITT : Ergebnisse

Geben wir zum Beispiel als Anfangsbestand den Wert 500 ein, und als Zuwachsrate im ersten Jahr den Wert 4 % und schließlich die Angabe, daß der Prozeß 8 Jahre dauern soll, so erhalten wir :

NACH ... JAHREN	BESTAND	ZUWACHS (%)
1	500	4
2	530.4	2
3	537.47	1.33
4	542.84	1
5	547.18	.8
6	550.83	.67
7	553.98	.57
8	556.75	.5

NOCHMAL ? (J/N) ?

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 4 : Umweltverschmutzung	Seite 223
-----------------------	--	--------------

6.4 Umweltverschmutzung

① SCHRITT : Vorstellung des Problems

Wachsende Populationen hemmen in der Regel ihre weitere Entwicklung selbst, nämlich dadurch, daß sie ihre Lebensbedingungen in zunehmendem Maße belasten.

Typisches Beispiel ist die zunehmende Umweltverschmutzung während der Industrialisierungsphasen.

Also auch hier tritt eine "Wachstumsbremse" zutage - ähnlich wie im vorangegangenen Abschnitt - die das ungebremste Wachstum, wie wir es in Abschnitt 6.2 kennengelernt haben, modifiziert.

Im Gegensatz aber nun zum Beispiel im vorangegangenen Abschnitt, wirkt diese "Bremse" nicht in Abhängigkeit von der Populationsgröße selbst, sondern indirekt in Abhängigkeit von einer Variablen, die ihrerseits proportional mit der Populationsgröße zusammenhängt.

2. SCHRITT : Problemanalyse

Die Population in der Periode 1 ergibt sich aus der der Periode ϕ jetzt nicht mehr als

$$X_1 = X_\phi + X_\phi * R/1\phi\phi$$

sondern als :

$$X_1 = X_\phi + X_\phi * (R/1\phi\phi - P*B\phi)$$

Dabei ist $B\phi$ die Umweltbelastung zum Zeitpunkt ϕ , die über einen geeigneten Parameter P (z.B. in der Größenordnung $R = \phi.\phi 1$) die Populationszuwächse bremst.

Natürlich brauchen wir jetzt aber auch eine funktionale Beziehung, welche die Größe B_1 (Umweltbelastung in der Periode 1) in Abhängigkeit von $B\phi$ und $X\phi$ erklären kann.

Als Beispiel einer solchen Beziehung mag gelten :

$$B_1 = B\phi + A * X\phi$$

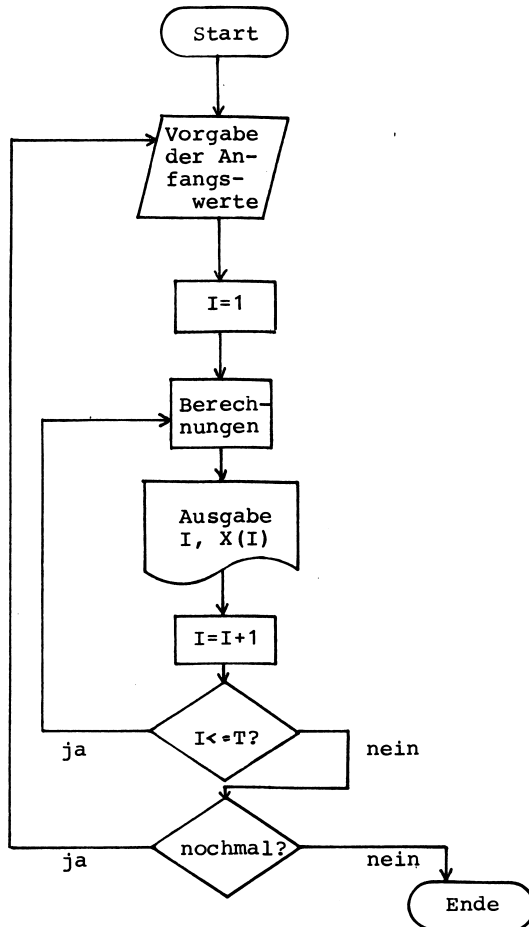
Von einer Belastungsgröße von z.B. $B\phi = \phi$ ausgehend, wächst die Belastung in Abhängigkeit von der erreichten Populationsgröße nach Maßgabe eines geeignet festzusetzenden Parameters A (z.B. $A = \phi.\phi 15$).

Somit gilt also der folgende Algorithmus :

$$(1) \quad B_1 = B\phi + A * X\phi$$

$$(2) \quad X_1 = X\phi + X\phi * (R/1\phi\phi - P * B_1)$$

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 4 : Umweltverschmutzung	Seite 226
-----------------------	--	--------------

4. SCHRITT : Programm

```

10 REM B3-UMWELT
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR TABELLARISCHEN DARSTELLUNG"
40 PRINTTAB(2)"EINES WACHSTUMSPROZESSES, DER DURCH"
50 PRINTTAB(2)"SEINE EIGENE DYNAMIK GEBREMST WIRD."
60 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"DIESES PROGRAMM BENDETIGT DIE FOLGENDEN"
90 PRINT"AUSGANGSINFORMATIONEN : ":PRINT
100 PRINT:INPUT"AUSGANGSBESTAND           : ";X0
105 PRINT:INPUT"WIEVIELE JAHRE           : ";T
110 PRINT:INPUT"JAEHRL. WACHSTUMSRATE IN % : ";R
120 PRINT:INPUT"FAKTOR UMWELTBELASTUNG    : ";P
130 PRINTCHR$(147)
140 PRINT"ZEIT  BESTAND  ZUWACHS %  BELASTUNG":PRINT:PRINT
150 I=1:A=0.015:B0=0
160 B1=B0+A*X0
170 X1=X0+X0*(R/100-P*B0)
175 IF X1<=0 THEN PRINT"POPULATION IST VERNICHTET":GOTO 210
180 Z=(X1-X0)/X0:Z=Z*100:Z=INT(Z*100+.5)/100
185 B1=INT(B1*100+.5)/100:X1=INT(X1*100+.5)/100
190 PRINTTAB(2)I;TAB(7)X1;TAB(17)Z;TAB(29)B1
200 I=I+1:IF I<=T THEN B0=B1:X0=X1:GOTO 160
210 PRINT:PRINT:INPUT"NOCHMAL ? (J/N) ";A$
220 IF A$="J"THEN PRINTCHR$(147):GOTO 80
230 PRINT:PRINT"ENDE DER AUSGABE":END

```

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 4 : Umweltverschmutzung	Seite 227
-----------------------	--	--------------

(5.) SCHRITT : Variablenliste

- A = Faktor populationsbedingte Umweltbelastung
- A\$ = Stringvariable (Ja, Nein)
- B ϕ = Umweltbelastung Anfang
- B1 = Umweltbelastung nächste Periode
- I = Laufindex
- P = Faktor Umweltbelastung
- R = Jährliche Wachstumsrate
- T = Zeit
- X ϕ = Population Anfang
- X1 = Population nächste Periode
- Z = Populationszuwachs.

(6.) SCHRITT : Programmbeschreibung

- Satz 1 ϕ -7 ϕ : Überschrift und Erläuterungen
- Satz 8 ϕ -12 ϕ : Anforderung der Inputinformationen
- Satz 13 ϕ -14 ϕ : Löschen des Bildschirms und Ausgabe einer Tabellenüberschrift
- Satz 15 ϕ : Vorgabe weiterer Ausgangsparameter
- Satz 16 ϕ -17 ϕ : Berechnungen für die folgende Periode
- Satz 175 : Wenn die neu berechnete Population kleiner oder gleich null wird, ist eine

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 4 : Umweltverschmutzung	Seite 228
<p style="text-align: center;">entsprechende Meldung auszugeben und Sprung zum Satz 21φ</p> <p>Satz 18φ-185 : Weitere notwendige Berechnungen</p> <p>Satz 19φ : Ausgabe der Periodenergebnisse</p> <p>Satz 2φφ : Erhöhung des Laufindex I</p> <p style="text-align: center;">solange I kleiner als T bleibt, wird Xφ mit X1 und Bφ mit B1 belegt und das Programm kehrt zurück zum Satz 16φ</p> <p>Satz 21φ-23φ : Abfrage, ob eine weitere Berechnung gewünscht wird (wenn ja, Löschen des Bildschirms und zurück zu Satz 8φ) und Beendigung des Programms</p>		

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 4 : Umweltverschmutzung	Seite 229
-----------------------	--	--------------

(7) SCHRITT : Ergebnisse

Geben wir als Ausgangsbestand z.B. den Wert 1000, als Zeithorizont 8 Jahre, als jährliche Wachstumsrate 5 % und als Faktor der Umweltbelastung z.B. den Wert 0.01 ein, so erhalten wir die folgenden Angaben :

ZEIT	BESTAND	ZUWACHS %	BELASTUNG
1	1050	5	15
2	945	-10	30.75
3	701.66	-25.75	44.93
4	421.49	-39.93	55.45
5	208.85	-50.45	61.77
6	90.29	-56.77	64.9
7	36.21	-59.9	66.25
8	14.03	-61.25	66.79

NOCHMAL ? (J/N) ?

Voß C64/ Schule	Kapitel 6 : Biologie/Ökologie Abschnitt 4 : Umweltverschmutzung	Seite 230

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 1 : Vorbemerkung	Seite 231
-----------------------	---	--------------



Kapitel 7 : Erdkunde/Geschichte
=====

7.1 Vorbemerkung

Es können in diesem Kapitel ähnliche Programme erstellt werden wie im Kapitel "Sprachen", also vor allem Test- und Abfrageprogramme. Auch hier werden wir uns dabei auf exemplarische Datenbestände beschränken, d.h. es wird nur gezeigt, wie die Programme aufgebaut werden müssen und wie sie dann funktionieren.

Sollen sie praktisch genutzt werden, müssen erst, wie unschwer zu sehen ist, die kompletten Datenbestände erzeugt und eingegeben werden. Dazu müssen die DATA-Statements ergänzt und die jeweilige Vorgabe von N entsprechend geändert werden.

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 2 : Jahreszahlen	Seite 232
-----------------------	---	--------------

7.2 Historische Jahreszahlen

① SCHRITT : Vorstellung des Problems

Es soll ein Programm vorgestellt werden, welches dazu benutzt werden kann, das Erlernen historischer Jahreszahlen zu erleichtern und die erlernten Daten zu trainieren.

Zu diesem Zweck stellt der Rechner dem Benutzer historische Ereignisse vor und verlangt von ihm, daß er die korrekte Jahreszahl dieses geschichtlichen Ereignisses angibt. Es bleibt dabei dem Benutzer überlassen zu entscheiden, wieviele Fragen er sich pro Testserie vom Rechner stellen lassen möchte.

Gibt der Benutzer die korrekte Antwort, so reagiert der Rechner mit Lob; erhält er hingegen keine korrekte Antwort, so soll er den Benutzer fragen, ob dieser noch einen Versuch wagen möchte. Wird dies abgelehnt, soll das Programm dafür sorgen, daß die korrekte Antwort auf dem Bildschirm erscheint.

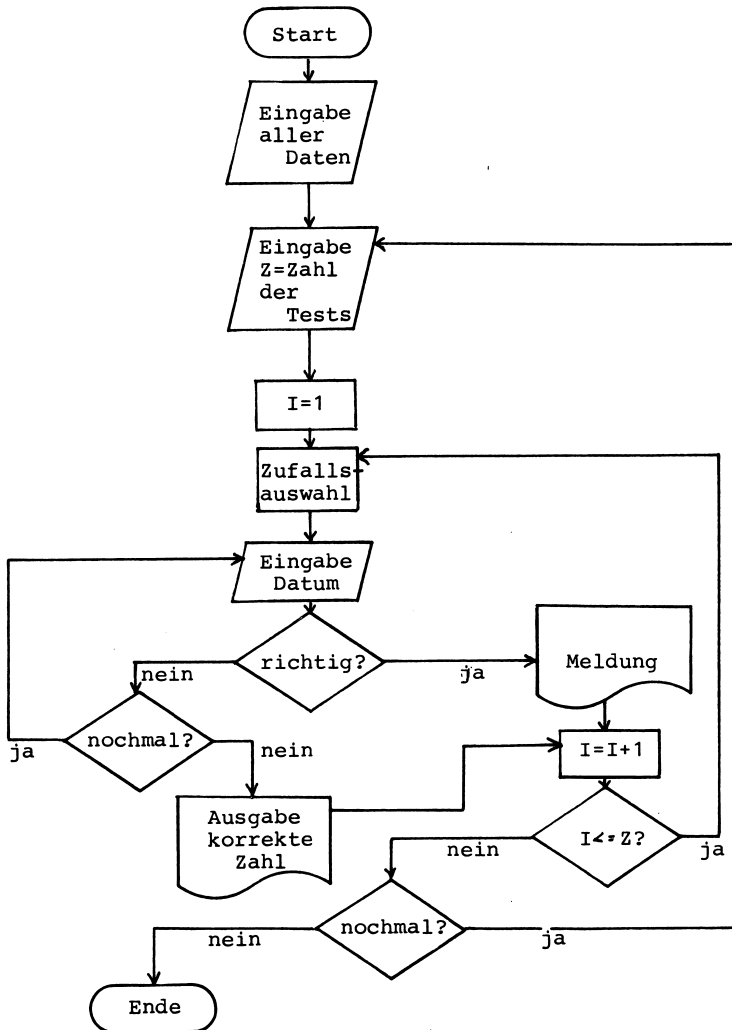
Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 2 : Jahreszahlen	Seite 233
-----------------------	---	--------------

②. SCHRITT : Problemanalyse

Die Vorstellung des Problems hat schon hinreichend verdeutlicht, wie bei diesem Programm im einzelnen vorzugehen ist :

Wir benötigen einen Informationseingabeteil, der Benutzer muß gefragt werden, wieviele Abfragen er über sich ergehen lassen möchte und das Programm muß die jeweils gegebenen Antworten prüfen. Desweiteren tauchen keine besonderen Schwierigkeiten auf.

3. SCHRITT : Flußdiagramm



④ SCHRITT : Programm

```
10 REM E1-JAHRESZAHLEN
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUM ABFRAGEN HISTORISCHER DATEN"
40 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
50 PRINT:PRINT:PRINT
60 PRINT"DIESES PROGRAMM BEZIEHT SICH NUR AUF 10"
70 PRINT"AUSGEWAELHTE HISTORISCHE DATEN (SIEHE"
80 PRINTTAB(5)"DATA-STATEMENTS IN 500 FF).":PRINT
90 PRINT"SO LLEEN MEHR DATEN VERWENDET WERDEN, SO"
100 PRINT"MUESSEN DIE DATA ERGAENZT UND N IN SATZ"
110 PRINTTAB(7)"150 VERAENDERT WERDEN."
120 PRINT:PRINT:PRINT
130 PRINTTAB(5)"BITTE EINE TASTE DRUECKEN !"
140 GET A$:IF A$=""THEN 140
150 N=10
155 DIM J$(N),S$(N)
160 PRINTCHR$(147)
170 FOR I=1TON:READJ$(I),S$(I):NEXTI
180 PRINT"WIEVIELE ABFRAGEN WERDEN GEWUENSCHT ?"
190 PRINT:INPUT"BITTE ZAHL ANGEBEN : ";Z
200 FOR I=1 TO Z
202 R=INT(RND(1)*N+1)
205 PRINTCHR$(147)
210 PRINT:PRINT"WANN WAR : ":PRINT
230 PRINT" ";S$(R);:PRINT" "
240 PRINT:PRINT"IM JAHR : ";:PRINT" ";:INPUT K$
250 IF K$=J$(R) THEN PRINT:PRINTTAB(9)"P R I M A";
:PRINT" ":GOTO 300
260 PRINT:PRINT"DAS WAR LEIDER NICHT RICHTIG !":PRINT" "
270 PRINT:PRINT"NOCH EIN VERSUCH ?":INPUT (J/N) ";A$
280 IF A$="J" THEN 205
290 PRINT:PRINT"DIE RICHTIGE JAHRESZAHL LAUTET : ";J$(R)
300 FORII=1TO2000:NEXTII:NEXTI
310 PRINT:PRINT:PRINT"NOCH EIN TEST ?":INPUT (J/N) ";A$
320 IF A$="J" THEN PRINTCHR$(147):GOTO 180
330 PRINT:PRINT"ENDE":END
500 DATA 1939,BEGINN DES ZWEITEN WELTKRIEGS
510 DATA 800,KAISERKROENUNG VON KARL DEM GROSSEN
520 DATA 1949,GRUENDUNG DER BUNDESREPUBLIK DEUTSCHLAND
530 DATA 9,SCHLACHT IM TEUTOBURGER WALD
540 DATA 1832,HAMBACHER FEST
550 DATA 1871,GRUENDUNG DES DEUTSCHEN KAISERREICHS
560 DATA 1890,ERFINDUNG DER LOCKKARTE
570 DATA 1812,NAPOLEON VOR MOSKAU
580 DATA 1763,FRIEDE VON HUBERTUSBURG
590 DATA 1495,REICHSTAG VON WORMS
```

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 2 : Jahreszahlen	Seite 236
-----------------------	---	--------------

5. SCHRITT : Variablenliste

A\$ = Stringvariable für Antworten
I = Laufindex
J\$ = Jahreszahlen
N = Anzahl der Datensätze
R = Zufallszahl
S\$ = Historische Sachverhalte
Z = Anzahl der Tests

6. SCHRITT : programmbeschreibung

Satz 10-12 : Überschrift und Erläuterungen

Satz 13-14 : Warten

Satz 15-17 : Vorgabe der Anzahl der Datensätze, Dimensionieren, Löschen des Bildschirms und Einlesen der Daten

Satz 18-19 : Anforderung der Zahl der Tests

Satz 20-30 : Tests :

Satz 22-24 : Zufallsauswahl eines historischen Ereignisses und Anforderung der Jahreszahl als Antwort des Benutzers

Satz 25 : Falls korrekt, Meldung und weiter bei 30

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 2 : Jahreszahlen	Seite 237
-----------------------	---	--------------

Satz 26φ-28φ : Falls nicht korrekt, Meldung und Frage, ob noch ein Versuch (wenn ja, zurück nach 2φ5)

Satz 29φ : Wenn nein, korrekte Ausgabe

Satz 3φφ : Warteschleife und dann zurück nach Satz 2φφ.

Satz 31φ-33φ : Frage, ob noch ein Testlauf gewünscht;

wenn ja, zurück zu Satz 18φ nach Löschen des Bildschirms

wenn nein, Ende des Programms

Satz 5φφ-59φ : Daten

7. SCHRITT : Ergebnisse

Die Ergebnisse brauchen hier nicht vorgestellt zu werden. Am sinnvollsten ist es, wenn der Leser unter Beachtung der Inhalte der DATA-Statements das Programm einfach ausprobiert und es dann gemäß seinen eigenen Wünsche erweitert.

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 3 : Länder	Seite 238
-----------------------	---	--------------

7.3 Die Hauptstädte der Länder

① SCHRITT : Vorstellung des Problems

Auch das Programm, das in diesem Abschnitt vorgestellt werden soll, ist ein sog. Trainingsprogramm. Der Benutzer soll auf die Fragen des Rechners Antwort geben. Gibt der Rechner zufällig ausgewählte Bundesländer vor, soll der Benutzer die Hauptstädte dieser Länder eingeben; gibt der Rechner hingegen Hauptstädte vor, soll der Benutzer die dazugehörigen Bundesländer nennen.

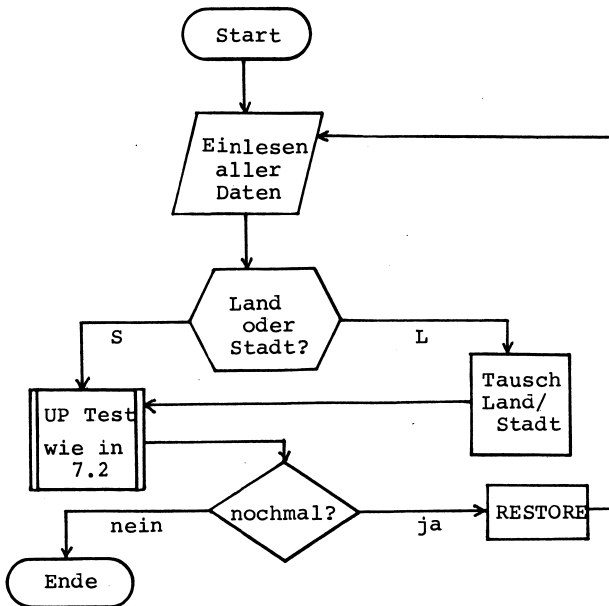
Insoweit bieten sich hier also keine neuen Probleme, so daß wir uns hier sehr kurz fassen können. Auch dieses Programm hat also eher exemplarischen Charakter :

Ohne Schwierigkeiten kann der Benutzer andere Sachverhalte in den DATA-Statements dieses Programms erfassen und gelangt auf diese Weise zu einem generell einsetzbaren Abfrageprogramm, das für die verschiedensten Schulfächer verwendet werden könnte.

2. SCHRITT : Problemanalyse

Es wurde schon erwähnt, daß hier keine neuen Probleme auftauchen, so daß dieser Schritt entsprechend kurz gehalten werden kann. Entsprechende Fragestellungen wurden in vorangegangenen Beispielen schon hinreichend diskutiert.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 3 : Länder	Seite 240
-----------------------	---	--------------

4. SCHRITT : Programm

```

10 REM E2-LAENDER
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR LERNKONTROLLE IM ERDKUNDE-"
40 PRINTTAB(14)"UNTERRICHT."
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT"DIESES PROGRAMM FORDERT NACH ANGABE DER"
80 PRINT"DEUTSCHEN BUNDESLAENDER DEREN HAUPT-"
90 PRINT"STAEDTE AN UND UMGEKEHRT.":PRINT:PRINT
100 PRINT"BEI ANDEREN VORGABEN VON DATEN IN DEN"
110 PRINT"STATEMENTS 500 FF. UND AENDERUNG VON N"
120 PRINT"IN SATZ 160 KANN DAS PROGRAMM AUCH FUER"
130 PRINT"ANDERE ZWECKE VERWENDET WERDEN."
140 PRINT:PRINT:PRINTTAB(6)"BITTE EINE TASTE DRUECKEN"
150 GET A$:IF A$="" THEN 150
160 N=11
170 DIM L$(N),S$(N)
180 FOR I=1TON:READL$(I),S$(I):NEXT I
190 PRINTCHR$(147)
200 PRINT"SOLLEN LAENDER ODER HAUPTSTAEDTE VORGE-"
210 INPUT"GEBEN WERDEN (L/S) ":A$
215 C$="HAUPTSTADT"
220 IF A$="S" THEN C$="LAND":GOTO 250
230 GOSUB 1000:REM TEST
240 GOTO 270
250 FOR I=1TON:H$=L$(I):L$(I)=S$(I):S$(I)=H$:NEXT I
260 GOSUB 1000:REM TEST
270 PRINT:PRINT:INPUT"NOCHMAL ? (J/N) ":A$
280 IF A$="J"THEN PRINTCHR$(147):RESTORE:GOTO 180
290 PRINT:PRINT"ENDE":END
500 DATA SCHLESWIG-HOLSTEIN,KIEL,HAMBURG,HAMBURG,BREMEN,
      BREMEN
510 DATA BERLIN,BERLIN,NIEDERSACHSEN,HANNOVER
520 DATA NORDRHEIN-WESTFALEN,DUESSELDORF,HESSEN,WIESBADEN
530 DATA RHEINLAND-PFALZ,MAINZ,SAARLAND,SAARBRUECKEN,BAYERN,
      MUENCHEN

```

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 3 : Länder	Seite 241
<pre> 540 DATA BADEN-WUERTTEMBERG,STUTTGART 1000 REM UP TEST 1010 PRINTCHR\$(147) 1020 PRINT:INPUT"WIEVIELE ABFRAGEN : ";Z 1030 FOR I=1 TO Z 1040 R=INT(RND(1)*N+1) 1050 PRINT:PRINT"┌";L\$(R):PRINT:PRINT:PRINT"┐";C\$;" : "; :PRINT:PRINT:INPUT X\$ 1060 IF X\$=S\$(R) THEN PRINT:PRINT TAB(10)"K O R R E K T" :PRINT"┐":GOTO 1100 1070 PRINT:PRINTTAB(10)"#FEHLER!":PRINT"┐" 1080 PRINT:PRINT"DIE RICHTIGE ANTWORT LAUTET : " 1090 PRINT:PRINTTAB(10)"┌";S\$(R):PRINT"┐" 1100 NEXT I 1110 RETURN </pre>		

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 3 : Länder	Seite 242
-----------------------	---	--------------

5. SCHRITT : Variablenliste

A\$ = Stringvariable (Ja,Nein)
C\$ = Hilfsvariable, die, je nach Auswahl, mit "Stadt"
oder "Land" belegt wird
I = Laufindex
L\$ = Länder
N = Anzahl der Datensätze
S\$ = Städte

6. SCHRITT : Programmbeschreibung

Satz 10-13 : Überschrift und Erläuterungen
Satz 14-15 : Warten
Satz 16-18 : Einlesen des gesamten Datenbestands
Satz 19-21 : Abfrage, ob Länder oder Hauptstädte
vorgegeben werden sollen
Satz 22 : werden Städte gewählt, so wird die
Hilfsvariable C\$ mit "Land" belegt
und es erfolgt ein Sprung nach 25
Satz 23 : Sprung ins Unterprogramm 1000, in
dem der eigentliche Abfragetest durch-
geführt wird

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 3 : Länder	Seite 243
	<p>Satz 24φ : Sprung zum Satz 27φ</p> <p>Satz 25φ : Austausch von Ländern und Städten, damit bei anderer Wahl in 19φ-21φ trotzdem wieder das gleiche Unterprogramm verwendet werden kann</p> <p>Satz 26φ : Wie Satz 23φ</p> <p>Satz 27φ-29φ : Abfrage, ob noch eine weitere Testserie gewünscht wird (wenn ja, Löschen des Bildschirms und zurück zu Satz 18φ, nachdem der Datenbestand restauriert wurde - dies ist notwendig, um den eventuellen Tausch in 25φ wieder rückgängig zu machen, um also wieder die Ausgangsbedingungen herzustellen); andernfalls Ende des Programms</p> <p>Satz 5φφ-54φ : Ausgangsdaten</p> <p>Satz 1φφφ-111φ : Unterprogramm Test :</p> <p style="padding-left: 40px;">Programm zur Durchführung der gewünschten Abfragen; die Einzelheiten entsprechen im Prinzip denen in Abschnitt 7.2, so daß wir hier auf eine detailliertere Beschreibung dieses Unterprogramms verzichten können</p>	

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 3 : Länder	Seite 244
-----------------------	---	--------------

7. SCHRITT : Ergebnisse

Auch bei diesem Programm ist es nicht erforderlich, Ergebnisausdrucke vorzustellen. Die Durchsicht des Programms oder besser noch das Ausprobieren zeigt sehr deutlich, was im einzelnen geschieht.

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 4 : Bevölkerung	Seite 245
-----------------------	--	--------------

7.4 Die Bevölkerungsentwicklung in verschiedenen Nationen

①. SCHRITT : Vorstellung des Problems

Es soll in diesem Abschnitt ein Programm vorgestellt werden, welches für verschiedene Länder dieser Erde die Bevölkerungsentwicklung bis zum Jahre 2000 prognostiziert.

Als zu betrachtende Länder bzw. Ländergruppen haben wir ausgewählt :

Welt (als alle Länder umfassend)

Industrieländer

Entwicklungsländer

VR China

Indien

Japan

USA

UdSSR

Bundesrepublik Deutschland

DDR.

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 4 : Bevölkerung	Seite 246
-----------------------	--	--------------

② SCHRITT : Problemanalyse

Die geplanten Bevölkerungsprognosen machen es erforderlich, daß man sich für eine bestimmte Prognosemethode entscheidet.

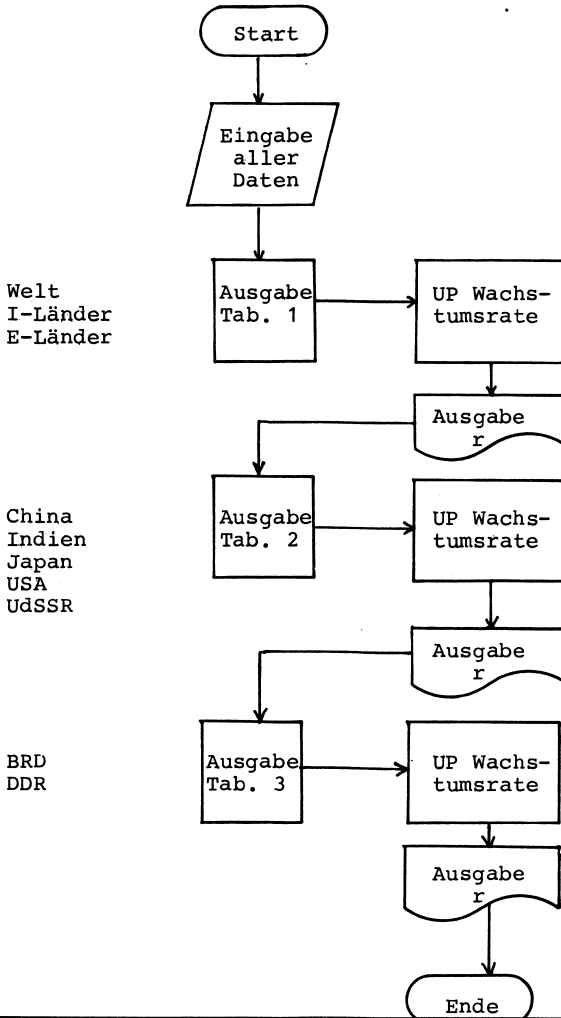
Es gibt eine große Zahl verschiedener Methoden, die im einzelnen nicht hier diskutiert werden können.

Wir haben für das folgende Programm die Werte übernommen die sich bei den Modellprognosen ergeben haben, wie sie in dem Bericht GLOBAL 2000 (Washington 1980) veröffentlicht wurden.

Aus diesen Daten berechnen wir zusätzlich die jahresdurchschnittlichen Wachstumsraten, um damit durch das Programm eine zusätzliche Information bereitstellen zu lassen.

Unter programmlogischen Gesichtspunkten tauchen keine besonderen Schwierigkeiten auf, wenn man einmal davon absieht, daß wegen der Fülle der Ergebnisse an mehreren Stellen Programmunterbrechungen erforderlich sind.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 4 : Bevölkerung	Seite 249
<pre> 385 PRINT:PRINT 390 GOSUB 1000:REM WARTEN 400 PRINTCHR\$(147) 410 PRINT" JAHR BUNDESREPUBLIK DDR":PRINT" " 420 GOSUB 1200:REM STRICH 430 FOR I=1 TO 6 440 PRINTJ(I);TAB(12)B(9,I);TAB(24)B(10,I):PRINT 450 NEXT I 460 GOSUB 1200:REM STRICH 470 FOR L=9 TO 10 480 X=B(L,1):Y=B(L,6) 490 GOSUB 1100:REM WACHSTUMSRATEN 492 NEXT L 494 PRINT" %":PRINT" ";PRINTTAB(12)WR(9);TAB(24)WR(10) 496 PRINT:PRINT"ENDE":END 500 DATA 1975,1980,1985,1990,1995,2000 505 REM WELT 510 DATA 4134,4549,5013,5545,6143,6798 515 REM INDUSTRIELAENDER 520 DATA 1131,1174,1224,1276,1327,1377 525 REM ENTWICKLUNGSLAENDER 530 DATA 3003,3375,3789,4269,4816,5420 535 REM VR CHINA 540 DATA 978,1071,1151,1241,1348,1468 545 REM INDIEN 550 DATA 618,694,786,894,1013,1142 555 REM JAPAN 560 DATA 112,117,122,127,131,135 565 REM USA 570 DATA 214,222,235,248,260,270 575 REM UDSSR 580 DATA 254,268,282,296,310,323 585 REM BRD 590 DATA 61.8,61.7,60.0,58.6,57.8,56.2 595 REM DDR 600 DATA 16.8,16.7,16.6,16.4,16.2,16.1 610 DATA "WELT","I-LAENDER","E-LAENDER","CHINA","INDIEN" 620 DATA "JAPAN","USA","UDSSR","BRD","DDR" 1000 REM UP WARTEN 1010 PRINTTAB(5)"BITTE EINE TASTE DRUECKEN !" 1020 GET A\$:IF A\$="" THEN 1020 1030 PRINTCHR\$(147) 1040 RETURN 1100 REM UP WACHSTUMSRATE 1110 R=EXP(LOG(Y/X)/25)-1 1120 R=R*100:R=INT(R*100+.5)/100 1130 WR(L)=R 1140 RETURN 1200 REM UP STRICH 1210 FORI=1TO40:PRINT"-":NEXT I 1220 PRINT:RETURN </pre>		

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 4 : Bevölkerung	Seite 250
-----------------------	--	--------------

5. SCHRITT : Variablenliste

B = Bevölkerung
I = Laufindex
J = Jahr
L = Laufindex
L\$ = Länder
WR = Wachstumsrate
X = Anfangswert (1975)
Y = Endwert (2000)

6. SCHRITT : Programmbeschreibung

Satz 10-60 : Überschrift
Satz 70-100 : Einlesen aller Ausgangsdaten
Satz 110 : Sprung ins UP 1000 : Warten
Satz 140-145 : Tabellenüberschrift
Satz 150 : Sprung ins UP 1200 : Unterstreichung
Satz 160-190 : Ausgabe der Tabellenwerte
Satz 200-230 : Bestimmung jahresdurchschnittlicher Zu-
wachsrate für die Länder L=1, L=2 und
L=3

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 4 : Bevölkerung	Seite 251
<p style="text-align: center;">Satz 21φ : Übernahme des Anfangs- und des Endwertes</p> <p style="text-align: center;">Satz 22φ : Sprung ins UP 11φφ : Berech- nung der Wachstumsrate</p> <p>Satz 24φ : Ausgabe der Wachstumsraten</p> <p>Satz 25φ : Warten</p> <p>Satz 26φ-39φ : Gleiche Vorgehensweise wie in Satz 14φ -25φ beschrieben, nun aber für die zwei- te Tabelle</p> <p>Satz 4φφ-494 : Entsprechend für die dritte Tabelle</p> <p>Satz 496 : Beendigung des Programms</p> <p>Satz 5φφ-62φ : Daten.</p> <p>Satz 1φφφ-1φ4φ: Unterprogramm Warten</p> <p>Satz 11φφ-114φ: Unterprogramm jahresdurchschnittliche Wachstumsrate (sie wird mit Hilfe der Logarithmenrechnung bestimmt, siehe Satz 111φ)</p> <p>Satz 12φφ-122φ: Unterprogramm Strich :</p> <p style="text-align: center;">Durch Aneinanderfügung von Minuszeichen wird auf dem Bildschirm ein Strich ge- zeichnet</p>		

7. SCHRITT : Ergebnisse

Dieses Programm erzeugt drei Tabellen nacheinander auf dem Bildschirm :

1. Tabelle (Angaben in Millionen)

Jahr	WELT	I-LAENDER	E-LAENDER
1975	4134	1131	3003
1980	4549	1174	3375
1985	5013	1224	3789
1990	5545	1276	4269
1995	6143	1327	4816
2000	6798	1377	5421
RATE %	2.01	.79	2.39

BITTE EINE TASTE DRUECKEN !

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 4 : Bevölkerung	Seite 253
-----------------------	--	--------------

2. Tabelle (Angaben in Millionen)

JAHR	CHINA	INDIEN	JAPAN	USA	UDSSR
1975	978	618	112	214	254
1980	1071	694	117	222	268
1985	1151	786	122	235	282
1990	1241	894	127	248	296
1995	1348	1013	131	260	310
2000	1468	1142	135	270	323
%	1.64	2.49	.75	.93	.97

BITTE EINE TASTE DRUECKEN !

3. Tabelle (Angaben in Millionen)

JAHR	BUNDESREPUBLIK	DDR
1975	61.8	16.8
1980	61.7	16.7
1985	60	16.6
1990	58.6	16.4
1995	57.8	16.2
2000	56.2	16.1
%	- .38	- .17

ENDE

Voß C64/ Schule	Kapitel 7 : Erdkunde/Geschichte Abschnitt 4 : Bevölkerung	Seite 254

8

Kapitel 8 : Die Wirtschaft
=====

8.1 Vorbemerkung

Das wesentliche Einsatzgebiet der Rechner war lange Zeit der wirtschaftliche und speziell der betriebliche und der kaufmännische Bereich. Man kann sagen, daß auch heute noch in diesen Bereichen Computer mit am häufigsten eingesetzt werden.

Die Problemstellungen, um die es nämlich in diesen Bereichen geht, eignen sich in der Regel sehr gut dazu, durch Rechner, heute auch durch Mikrocomputer, gelöst zu werden.

Einige der grundlegenden wirtschaftlichen Probleme, zumindest so weit Rechenprobleme betroffen sind, werden in der Regel auch im schulischen Unterricht behandelt; beispielsweise in Handelsschulen natürlich in starkem Maße; aber auch in anderen Schultypen verschließt man nicht die Augen vor derartigen Fragestellungen, wie sie in diesem Kapitel behandelt werden sollen.

Auch hier gilt aber, daß die ausgewählten Probleme nur stellvertretend stehen für viele andere, die allein schon aus Platzgründen nicht aufgegriffen werden.

8.2 Die Zinsrechnung

① SCHRITT : Vorstellung des Problems

Man stelle sich vor, jemand bringt einen bestimmten Kapitalbetrag zu einer Bank, die dafür eine jährliche Zinszahlung vorsieht, deren Höhe alternativ festsetzbar ist.

Nach jedem Jahr vermehrt sich also der Kapitalbestand um einen bestimmten Prozentsatz, wobei zu beachten ist, daß ab dem zweiten Jahr auch die schon angefallenen Zinsen wiederum verzinst werden (Zinseszinsrechnung).

Es soll nun ein Programm entwickelt werden, welches für einen beliebigen Ausgangsbetrag und für einen beliebigen prozentualen Zinssatz ausrechnet, wie hoch der Endbetrag nach einer beliebigen Anzahl von Jahren sein wird.

Ein derartiges Programm kann übrigens als Muster für die Simulation unterschiedlicher Wachstumsvorgänge benutzt werden.

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 2 : Zinsrechnung	Seite 257
-----------------------	--	--------------

(2) SCHRITT : Problemanalyse

Nennen wir den Ausgangsbetrag X_0 , den Zinssatz P (%) und die Dauer der Bankeinlage T (Jahre).

Dann gilt nach dem ersten Jahr, daß sich der Betrag X_1 ergibt zu :

$$X_1 = X_0 + X_0 * P/100 = X_0 * (1 + P/100)$$

Im zweiten Jahr ergibt sich entsprechend zum Jahresende:

$$\begin{aligned} X_2 &= X_1 + X_1 * P/100 = X_1 * (1 + P/100) \\ &= X_0 * (1 + P/100) * (1 + P/100) \\ &= X_0 * (1 + P/100)^2 \end{aligned}$$

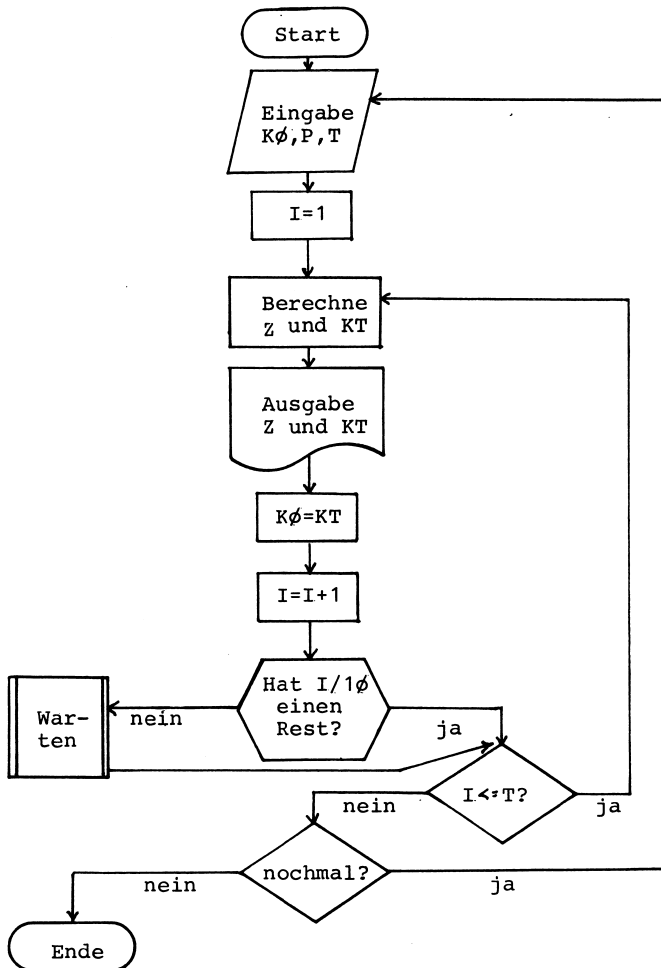
(Diese Beziehungen sind wir übrigens schon in einem früheren Kapitel begegnet).

Nach T Jahren ergibt sich demnach allgemein :

$$X_T = X_0 * (1 + P/100)^T$$

Mit Hilfe dieser Beziehung kann nun ein geeignetes Programm aufgebaut werden.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 2 : Zinsrechnung	Seite 259
-----------------------	--	--------------

4. SCHRITT : Programm

```

10 REM W1-ZINSRECHNUNG
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR BERECHNUNG VON ZINSEN BEI"
40 PRINT"JAEHLICHER ZINSZAHLUNG UND GEGEBENEM"
50 PRINTTAB(11)"AUSGANGSKAPITAL."
60 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT:PRINT
80 INPUT"AUSGANGSKAPITAL      : ";K0
90 PRINT:INPUT"JAEHRL.ZINSSATZ IN %: ";P
100 PRINT:INPUT"ANZAHL DER JAHRE      : ";T
110 PRINTCHR$(147)
120 PRINT"  JAHRE      ZINS      KAPITAL":PRINT"  "":PRINT
130 FOR I=1 TO T
140 KT=K0*(1 + P/100)
150 Z=K0*P/100
160 KT=K0+Z
170 Z=INT(Z*100+.5)/100
180 KT=INT(KT*100+.5)/100
190 PRINTTAB(2)I;TAB(8)Z;TAB(18)KT
200 K0=KT
210 IF I/10=INT(I/10) THEN GOSUB 500:REM WARTEN
220 NEXT I
230 PRINT:PRINT:PRINT"NOCH EINE BERECHNUNG ? (J/N)":INPUT A$
240 IF A$="J" THEN PRINTCHR$(147):GOTO 80
250 PRINT:PRINT:PRINT"ENDE DER BERECHNUNGEN":END
500 REM UP WARTEN
510 PRINT:PRINT:PRINTTAB(5)"BITTE EINE TASTE DRUECKEN !"
520 GETA$:IF A$=""THEN 520
530 PRINTCHR$(147)
540 PRINT"  JAHRE      ZINS      KAPITAL":PRINT"  "":PRINT
550 RETURN

```

VoB C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 2 : Zinsrechnung	Seite 260
-----------------------	--	--------------

⑤. SCHRITT : Variablenliste

I = Laufindex
 K ϕ = Kapitalbestand am Anfang
 KT = Kapitalbestand in den späteren Perioden
 P = Jährlicher Zinssatz in Prozent
 T = Dauer des Sparprozesses in Jahren
 z = Zinsbetrag pro Jahr

⑥. SCHRITT : Programmbeschreibung

Satz 1 ϕ -7 ϕ : Überschrift

Satz 8 ϕ -1 $\phi\phi$: Anforderung der Ausgangsdaten

Satz 11 ϕ -12 ϕ : Räumen des Bildschirms und Ausgabe einer Tabellenüberschrift

Satz 13 ϕ -22 ϕ : Berechnungen :

15 ϕ : Zinsbetrag

16 ϕ : Kapitalbetrag in der nächsten Periode

17 ϕ -18 ϕ : Rundung auf zwei Dezimalstellen

19 ϕ : Ausgabe

2 $\phi\phi$: Neubelegung von K ϕ mit KT, um in die nächste Periodenberechnung "einsteigen" zu können

21 ϕ : Nach je 1 ϕ Perioden Programmunter-

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 2 : Zinsrechnung	Seite 261
-----------------------	--	--------------

brechung durch Sprung ins UP 500.

Satz 230-250 : Beendigung des Programms, es sei denn,
es wird noch eine Berechnung gewünscht
(dann zurück zu Satz 80)

Satz 500-550 : Unterprogramm zum Warten, wenn der Bild-
schirm voll ist :

510 : Hinweis

520 : Warten, bis Tastendruck erfolgt

530 : Löschen des Bildschirms

540 : Erneute Ausgabe der Tabellenüber-
schrift

550 : Rücksprung

7. SCHRITT : Ergebnisse

Geben wir als Anfangskapitalbestand DM 100 ein, als
Zinssatz 5.5 % und als Zeitdauer 6 Jahre, dann ergibt
sich :

JAHR	ZINS	KAPITAL
1	5.5	105.5
2	5.8	111.3
3	6.12	117.42
4	6.46	123.88
5	6.81	130.69
6	7.19	137.88

NOCH EINE BERECHNUNG ? (J/N) ?

8.3 Hypothekentilgung

1. SCHRITT : Vorstellung des Problems

Unter den wirtschaftlichen Fragestellungen spielen diejenigen eine wichtige Rolle, die mit der Rückzahlung von Krediten oder Hypotheken zu tun haben. Geeignete Computerprogramme können die Rückzahlungsdauer unter unterschiedlichen Ausgangsbedingungen, wie z.B. Höhe der Verbindlichkeit, Zinshöhe und Rückzahlungsmodalitäten bestimmen.

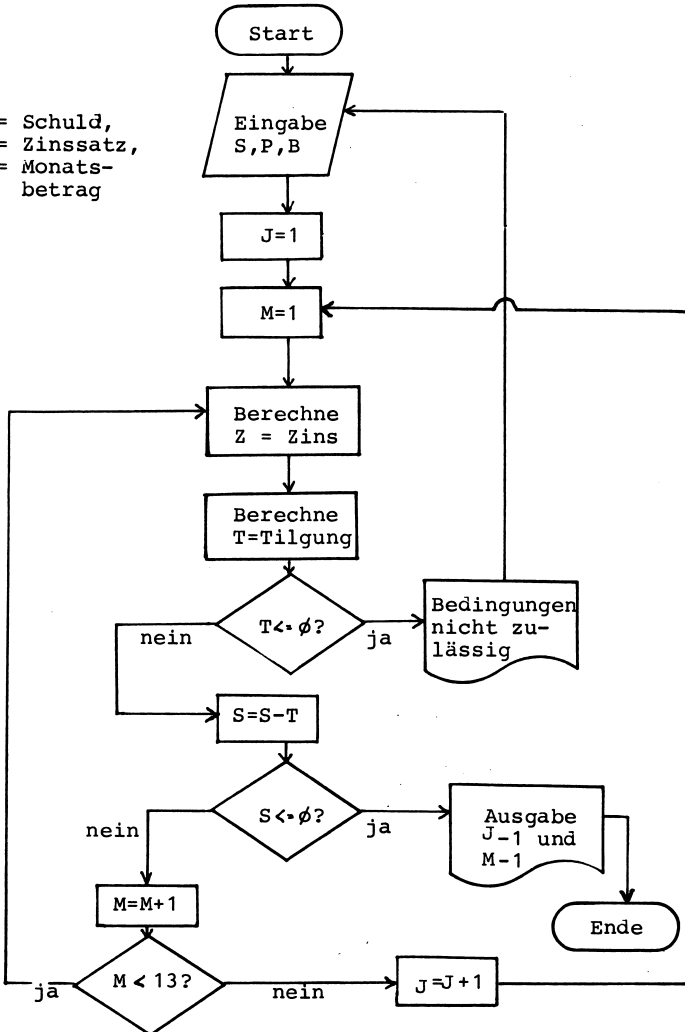
Es soll im folgenden ein Programm vorgestellt werden, welches für alternativ eingebbare Beträge die Tilgungsdauer berechnet, wenn alternative Zinssätze vorgegeben werden, bzw. wenn unterschiedlich hohe Rückzahlungsraten vereinbart werden.

Es ist leicht einzusehen, daß auch dieses Programm dann für generellere Fragestellungen dieser Art eingesetzt werden kann.

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 3 : Hypothek	Seite 263
<p style="text-align: center;"> 2, SCHRITT : Problemanalyse </p> <p> Zur Erläuterung der Problemlösung gehen wir von folgendem Beispiel aus : </p> <p> Jemand nimmt eine Hypothek in der Höhe von S DM auf und vereinbart mit der hypothekgewährenden Bank eine monatliche Rückzahlung von B DM. Von diesem Monatsbetrag B müssen zunächst die Schuldzinsen beglichen werden. Der Betrag, der nach Abzug der Zinsen übrig bleibt, kann zur Tilgung der Schuld, d.h. zu ihrer Verringerung verwendet werden - vorausgesetzt, es bleibt überhaupt ein positiver Rest übrig. </p> <p> Das zu entwickelnde Programm muß also Monat für Monat die anfallenden Zinsen berechnen und den Restbetrag, mit dem getilgt werden kann. Sollte dieser Restbetrag null oder negativ werden, so muß das Programm eine Meldung ausgeben, aus der erkenntlich wird, daß unter den gegebenen Bedingungen eine Verringerung der Schuld nicht möglich ist. </p> <p> Jahr für Jahr soll der aktuelle Schuldenstand ausgegeben werden und wenn der Schuldenstand null oder negativ geworden sein sollte, ist anzugeben, wie lange dieser Schuldentilgungsprozeß gedauert hat. </p>		

3. SCHRITT : Flußdiagramm

S = Schuld,
P = Zinssatz,
B = Monats-
betrag



VoB C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 3 : Hypothek	Seite 265
<p> 4. SCHRITT : Programm </p> <pre> 10 REM W2-HYPOTHEK 20 PRINTCHR\$(147) 30 PRINT"PROGRAMM ZUR BERECHNUNG DER DAUER EINER" 40 PRINTTAB(10)"HYPOTHEKENTILGUNG." 50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984" 60 PRINT:PRINT:PRINT:PRINT 70 PRINT:INPUT"MONATLICHE RATE : ";B 80 PRINT:INPUT"Z INSSATZ IN % : ";P 90 PRINT:INPUT"AKTUELLE HYP.SCHULD : ";S 100 PRINTCHR\$(147) 110 PRINT"JAHR SCHULD":PRINT"□" 120 FOR I=1 TO 20:PRINT"-";:NEXT I:PRINT 130 J=1 140 M=1 150 Z=((P/100)*S)/12:T=B-Z 160 IF T<=0 THEN 250 170 S=INT((S-T)*100+.5)/100 180 IF S<=0 THEN 300 190 M=M+1:IF M<13 THEN 150 200 PRINTTAB(2)J:TAB(8)S 210 J=J+1:GOTO 140 250 PRINT:PRINT:PRINT"DIE KONDITIONEN SIND DERART, DASS NICHT" 260 PRINT"GETILGT WERDEN KANN." 270 PRINT:PRINT"BITTE NEUE EINGABE":PRINT:GOTO 70 300 PRINT:PRINT:PRINT"TILGUNSDAUER :":PRINT 310 PRINTJ-1;" JAHRE UND ";M-1;" MONATE." 320 PRINT:PRINT:PRINT"NOCH EINE BERECHNUNG ? (J/N) ":INPUTA\$ 330 IF A\$="J" THEN PRINTCHR\$(147):GOTO 70 340 PRINT:PRINT:PRINT"ENDE DER BERECHNUNGEN":END </pre>		

5. SCHRITT : Variablenliste

A\$ = Stringvariable (Ja, Nein)
B = Monatlicher Rückzahlungsbetrag
I = Laufindex
J = Jahreszähler
M = Monatszähler
P = Jährlicher Schulzinssatz
T = Monatlicher Tilgungsbetrag
Z = Monatlicher Zinsbetrag

6. SCHRITT : Programmbeschreibung

Satz 10-60 : Überschrift
Satz 70-90 : Vorgabe der Ausgangsinformationen
Satz 100-120 : Ausgabe einer Tabellenüberschrift
Satz 130-140 : Vorgabe eines Jahreszählers (J) und
eines Monatszählers M
Satz 150-170 : Berechnung von monatlichem Zins, mo-
natlicher Tilgung und neuer Schuld
Satz 160 : Wenn nicht getilgt werden kann, dann
weiter bei 250
Satz 180 : Ist die Schuld getilgt, dann weiter
bei 300

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 3 : Hypothek	Seite 267
	<p>Satz 19φ : Nächster Monat falls M kleiner als 13 ist, weiter bei Satz 15φ, andernfalls bei Satz 2φφ</p> <p>Satz 2φφ : Ausgabe der aktuellen Schuld</p> <p>Satz 21φ : Nächstes Jahr und weiter bei Satz 14φ</p> <p>Satz 25φ-27φ : Meldung bei unzulässigen Tilgungskon- ditionen und zurück zu Satz 7φ</p> <p>Satz 3φφ-31φ : Ausgabe der Tilgungsdauer in Jahren und Monaten (der Leser überlege, wa- rum dabei von J und von M jeweils der Wert 1 abgezogen werden muß)</p> <p>Satz 32φ-34φ : Beendigung des Programms, es sei denn es wird eine erneute Berechnung ge- wünscht (dann nach Löschen des Bild- schirms zurück zum Satz 7φ)</p>	

7. SCHRITT : Ergebnisse

Geben wir bei den Anforderungen des Rechners beispielsweise ein :

MONATLICHE RATE : ? 500
ZINSSATZ IN % : ? 6.5
AKTUELLE HYP: SCHULD : ? 40000

so erhalten wir während des Programmablaufs :

JAHR	SCHULD
1	36496.86
2	32759.09
3	28771.02
4	24515.85
5	199975.7
6	15131.47
7	9962.84
8	4448.06

TILGUNGSDAUER :

8 JAHRE UND 9 MONATE.

NOCH EINE BERECHNUNG ? (J/N) ?

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 4 : Durchschnitt	Seite 269
-----------------------	--	--------------

8.4 Arithmetisches Mittel

①. SCHRITT : Vorstellung des Problems

Bei vielen wirtschaftlichen Fragestellungen ist es erforderlich, aus einer gegebenen Anzahl von Daten einen Durchschnitt auszurechnen, wobei üblicherweise das arithmetische Mittel benutzt wird.

Das folgende Programm soll für eine beliebige Anzahl von Daten nach Aufnahme dieser Werte diese Rechenprozedur durchführen.

②. SCHRITT : Problemanalyse

Bei der Problemanalyse treten keine besonderen Schwierigkeiten auf.

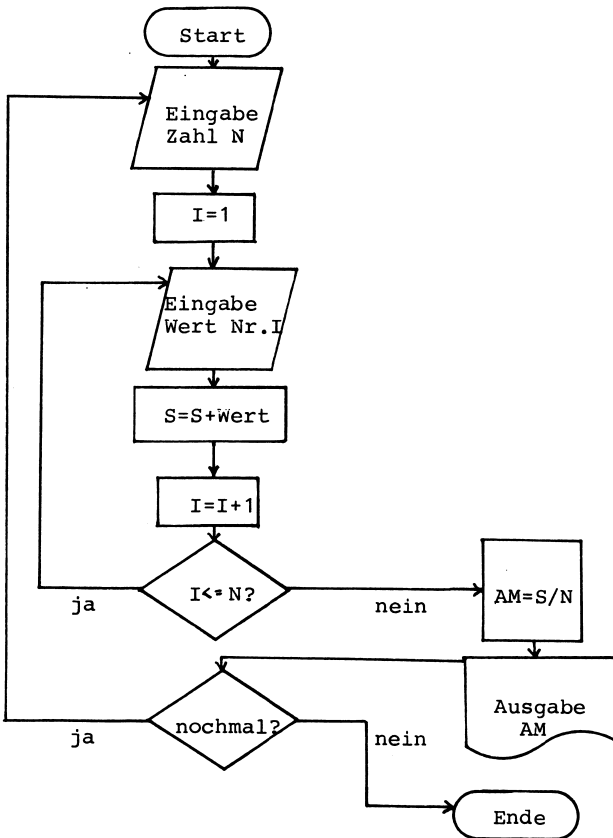
Es muß zunächst dafür gesorgt werden, daß eine offene Anzahl von Daten eingegeben werden kann, weshalb dem Rechner als erstes mitgeteilt werden muß, wieviele Werte es überhaupt sein werden.

Bei der Eingabe der Werte können diese gleich aufaddiert werden (das arithmetische Mittel ist ja die Summe aller Werte geteilt durch ihre Anzahl). Die Division der entstehenden Summe durch die Anzahl der Werte kann

dann als Ergebnis des Programms ausgegeben werden.

Soll eine weitere Berechnung durchgeführt werden, verzweigt das Programm wieder an den Anfang, andernfalls ist es beendet.

3. SCHRITT : Flußdiagramm



Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 4 : Durchschnitt	Seite 271
-----------------------	--	--------------

④. SCHRITT : Programm

```

10 REM W3-MITTELWERT
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR BERECHNUNG EINES ARITHME-"
40 PRINTTAB(11)"TISCHEN MITTELS."
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT:PRINT
70 INPUT"WIEVIELE WERTE : ";N
80 PRINTCHR$(147):S=0
90 FOR I=1 TO N
100 PRINTI:". WERT : ";:INPUT X
110 S=S+X
120 NEXT I
130 AM=S/N
150 PRINT:PRINT:PRINT"■ MITTELWERT = ";AM:PRINT"□"
160 PRINT:PRINT:PRINT"NOCH EINE BERECHNUNG ? (J/N)";:INPUT A$
170 IF A$="J" THEN PRINTCHR$(147):GOTO 70
180 PRINT:PRINT:PRINT"ENDE DER BERECHNUNGEN":END

```

⑤. SCHRITT : Variablenliste

AM = Arithmetisches Mittel
A\$ = Stringvariable (Ja, Nein)
I = Laufindex
N = Anzahl der zu mittelnden Werte
S = Summe der zu mittelnden Werte
X = Wert, der in die Mittelwertberechnung eingeht

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 4 : Durchschnitt	Seite 272
-----------------------	--	--------------

6. SCHRITT : Programmbeschreibung

Satz 1 ϕ -6 ϕ : Überschrift

Satz 7 ϕ : Angabe der Anzahl der zu mittelnden
Werte

Satz 8 ϕ : Löschen des Bildschirms und Vorgabe
eines Summenfeldes

Satz 9 ϕ -12 ϕ : Eingabe eines Wertes nach dem anderen
und Aufaddition aller Werte

Satz 13 ϕ -15 ϕ : Berechnung des arithmetischen Mittels
und Ausgabe des Ergebnisses

Satz 16 ϕ -18 ϕ : Abschluß des Programms, es sei denn, es
wird eine zusätzliche Berechnung ge-
wünscht

7. SCHRITT : Ergebnisse

Geben wir beispielsweise an, daß 3 Werte vorliegen, so
fordert das Programm genau drei Werte an.

Sind dies z.B. die Werte 5,7 und 9, so erhalten wir :

MITTELWERT = 7

NOCH EINE BERECHNUNG ? (J/N) ?

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 5 : Verteilung	Seite 273
-----------------------	--	--------------

8.5 Häufigkeitsverteilung

①. SCHRITT : Vorstellung des Problems

Bei der Auswertung größerer Datenbestände, für die sich ja der Einsatz von Computern ganz besonders gut eignet, besteht ein erster statistischer Auswertungsschritt häufig darin, eine sog. Häufigkeitsverteilung zu erstellen.

Es geht dabei darum, den Merkmalswerten (oder Merkmalswertklassen) einer Untersuchungsvariablen die Häufigkeit ihres Auftretens graphisch oder tabellarisch zuzuordnen.

Das folgende Programm wird sowohl eine graphische wie auch eine tabellarische Häufigkeitsverteilung erstellen, wobei uns als Datenmaterial die Altersverteilung in der Bundesrepublik Deutschland im Jahre 1980 dient.

Diese Altersverteilung der westdeutschen Bevölkerung stellt sich folgendermaßen dar :

Altersverteilung in der Bundesrepublik Deutschland
1980 :

Altersklasse	Anteil in %
unter 10	10.3
10 bis unter 20	16.4
20 bis unter 30	14.6
30 bis unter 40	13.4
40 bis unter 50	14.2
50 bis unter 60	11.9
60 bis unter 70	8.9
70 bis unter 80	7.8
80 bis unter 90	2.4
90 und mehr	0.2

2. SCHRITT : Problemanalyse

Bei der Problemanalyse gehen wir davon aus, daß uns die Daten wie in obiger Tabelle schon vorliegen, so daß diese Werte als Ausgangsmaterial dem zu entwickelnden Programm eingegeben werden können.

Sollte dies nicht der Fall sein, hätte man also beispielsweise individuelle Altersangaben in einer Fragebogenaktion o.ä. erhoben, so müßte in einem ersten Programmteil erst dafür gesorgt werden, daß die individuellen Angaben in die einzelnen, zunächst noch leeren, Altersklassen einsortiert werden (dies kann mit einer Folge von IF ... THEN-Statements geleistet wer-

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 5 : Verteilung	Seite 275
-----------------------	--	--------------

den, bei der jeder einzelne Alterswert überprüft wird).

Ist dies geschehen, dann könnten die Klassenbesetzungszahlen, die sich dann ergeben haben, durch Beziehung auf die Gesamtzahl der Angaben relativiert werden, so daß man zu Prozentanteilen gelangt, wie sie in obiger Tabelle schon vorgegeben sind.

Dieser Programmteil könnte also schematisch folgendermaßen aussehen :

```

100 INPUT "Nächstes Alter : ";X
110 IF X < 10 THEN Z1=Z1+ 1 : GOTO 100
120 IF X < 20 THEN Z2=Z2+ 1 : GOTO 100
...
usw.

200 R1=(Z1/N)*100
210 R2=(Z2/N)*100
...
usw.

```

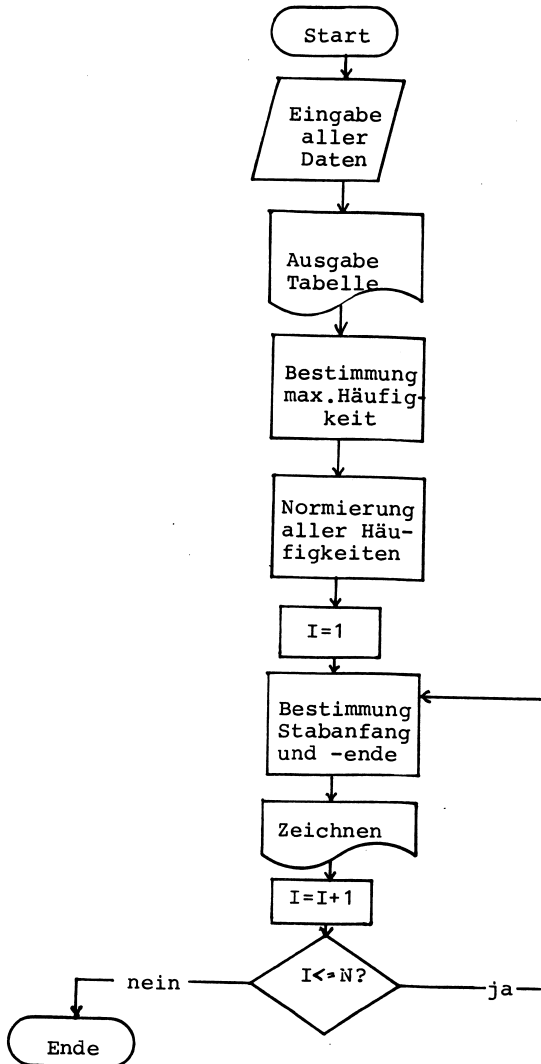
Hier in diesem Beispiel geht es aber lediglich darum, die schon vorgegebenen Werte zunächst tabellarisch darzustellen, was natürlich überhaupt keine programmtechnischen Schwierigkeiten verursacht.

Etwas komplizierter ist hingegen die graphische Darstellung dieser Häufigkeitsverteilung, bei der wir uns der Möglichkeiten bedienen wollen, über die im Kapitel "Physik" ausführlicher gesprochen wurde.

Um den Bildschirm für die graphische Darstellung optimal ausnutzen zu können, bestimmen wir zunächst aus dem vorliegenden Datenbestand die größte Häufigkeit und

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 5 : Verteilung	Seite 276
<p>rechnen alle anderen Häufigkeiten dann so um, daß die größte Häufigkeit in der graphischen Darstellung sich in der Vertikalen über 2ø Bildschirmzeilen erstreckt.</p> <p>Die Häufigkeiten selbst stellen wir als senkrechte Stäbe dar, die sich aus den schon mehrfach benutzten "Kügelchen" zusammensetzen. Um diese Stäbe maßstabgerecht zeichnen zu lassen, muß ihr jeweiliger Endpunkt bestimmt werden. Zusammen mit dem Anfangspunkt erhalten wir dann jeweils zwei Bildschirmadressen und es geht dann nur noch darum, alle Adressen zwischen diesen beiden (unter Einschluß derselben) mit "Kügelchen" zu belegen.</p>		

3. SCHRITT : Flußdiagramm



Voß
C64/
Schule

Kapitel 8 : Wirtschaft
Abschnitt 5 : Verteilung

Seite
278

4. SCHRITT : Programm

```
10 REM W5-VERTEILUNG
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR TABELLARISCHEN UND GRAPHI-
40 PRINT"SCHEN DARSTELLUNG EINER STATISTISCHEN"
50 PRINTTAB(8)"HAEUFIGKEITSVERTEILUNG."
60 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT:PRINT
80 PRINT"DARGESTELLT WIRD DIE ALTERSVERTEILUNG "
90 PRINT"IN DER BUNDESREPUBLIK DEUTSCHLAND 1980"
100 PRINT:PRINT:PRINT:PRINT
110 GOSUB 1000:REM WARTEN
120 DIM F(10),FT(10)
130 FORI=1 TO 10:READ F(I):NEXT I
140 PRINT"ALTER ANTEIL IN %":PRINT" "
150 FOR I=1 TO 35:PRINT"-":NEXT I:PRINT
160 FOR I=1 TO 10
165 A=(I-1)*10:B=I*10:IF I=10 THEN B=000
170 PRINT A;TAB(4)"BIS UNTER ";B;TAB(24)F(I)
180 NEXT I
190 PRINT:PRINT:PRINT:GOSUB 1000:REM WARTEN
200 FOR I=1866 TO 1896
210 POKE I,64:NEXT I
220 REM NORMIERUNG
230 FM=0
240 FORI=1TO10:IFF(I)>FM THEN FM=F(I)
250 NEXT I
260 FOR I=1 TO 10
270 FT(I)=F(I)*(20/FM):FT(I)=INT(FT(I)*100+.5)/100
280 NEXT I
290 REM ZEICHNEN
300 FOR I=1 TO 10
310 H1 = 21-FT(I):H1=INT(H1+.5)
320 S = I*3+3
330 A = 1024+S+H1*40
340 B = 1024+S+840
350 FOR Z=A TO B STEP 40
360 POKE Z+54272,2
370 POKE Z,81
380 NEXT Z
390 NEXT I
400 POKE 1910,49:POKE 1913,50:POKE 1916,51:POKE 1919,52:POKE 1922,53
410 POKE 1925,54:POKE 1928,55:POKE 1931,56:POKE 1934,57:POKE 1937,48
430 PRINT"ENDE":END
500 DATA 10.3,16.4,14.6,13.4,14.2,11.9,8.9,7.8,2.4,.2
1000 REM UP WARTEN
1010 PRINTTAB(5)"BITTE EINE TASTE DRUECKEN "
1020 GET A$:IF A$="" THEN 1020
1030 PRINTCHR$(147)
1040 RETURN
```

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 5 : Verteilung	Seite 279
-----------------------	--	--------------

⑤. SCHRITT : Variablenliste

- A = Anfang der Altersklasse
später : Anfangsadresse des zu zeichnenden Stabes
- A\$ = Stringvariable zur Antwortaufnahme
- B = Ende der Altersklasse
später : Endadresse des zu zeichnenden Stabes;
- F = Häufigkeit
- FM = maximale Häufigkeit
- FT = transformierte Häufigkeit
- H1 = Hilfsfeld zur Bestimmung einer Bildschirmzeile
- I = Laufindex
- S = Bildschirmspalte
- Z = Laufindex (Bildschirmzeile)

⑥. SCHRITT : Programmbeschreibung

- Satz 1 ϕ -1 $\phi\phi$: Überschrift
- Satz 11 ϕ : Warten
- Satz 12 ϕ -13 ϕ : Dimensionieren und Einlesen der Ausgangsdaten
- Satz 14 ϕ -15 ϕ : Ausgabe der Tabellenüberschrift
- Satz 16 ϕ -18 ϕ : Tabellenausgabe:
 - 165 : Bestimmung von Klassenanfangs- und Klassenendpunkt
 - 17 ϕ : Ausgabe

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 5 : Verteilung	Seite 280
	<p>Satz 19ϕ : Warten.</p> <p>Satz 20ϕ-21ϕ : Ausgabe einer waagrechten Achse für die graphische Darstellung</p> <p>Satz 22ϕ-25ϕ : Aufsuchen der größten Häufigkeit durch Vergleich aller Häufigkeiten mit der jeweils schon erreichten größten Häufigkeit (beim Wert null beginnend)</p> <p>Satz 26ϕ-28ϕ : Bestimmung transformierter Häufigkeiten durch Ausrichten der Häufigkeiten an der Maximalhäufigkeit</p> <p>Satz 29ϕ-39ϕ : Zeichnen von 1ϕ Stäben :</p> <p style="padding-left: 40px;">31ϕ : Bestimmung des Stabendes gemessen an der Bildschirm-Zeileneinteilung</p> <p style="padding-left: 40px;">32ϕ : Spalte, in der gezeichnet wird;</p> <p style="padding-left: 40px;">33ϕ : Anfangsadresse</p> <p style="padding-left: 40px;">34ϕ : Endadresse</p> <p style="padding-left: 40px;">35ϕ-38ϕ : Zeichnen des Stabes in roter Farbe</p> <p>Satz 40ϕ-41ϕ : Beschriftung der waagrechten Achse</p> <p>Satz 43ϕ : Ende des Programms</p> <p>Satz 50ϕ : Daten</p> <p>Satz 100ϕ-104ϕ: Unterprogramm Warten</p>	

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 5 : Verteilung	Seite 281
-----------------------	--	--------------

⑦. SCHRITT : Ergebnisse

Die Ergebnisse dieses Programms sollen hier nicht vorgestellt werden. Viel sinnvoller ist es, wenn der Leser das Programm in seinen Rechner eingibt, um dann im Vergleich der Bildschirmausgabe mit dem vorgestellten Programm nachzuvollziehen, was im einzelnen geschieht.

8.6 Die Wirtschaftskraft der Bundesländer

①. SCHRITT : Vorstellung des Problems

Von ganz anderer Art wie die vorangegangenen Programmbeispiele ist das folgende Programm. Hier geht es nämlich wieder, wie schon in manchen Kapiteln zuvor, darum, ein "Informationsprogramm" zu erstellen.

Es soll ein Programm entwickelt werden, welches für die Länder der Bundesrepublik Deutschland wichtige Wirtschaftsdaten auf Anforderung durch den Benutzer ausgibt.

Die Daten, die für diese Informationsbereitstellung vorgegeben werden, sind die folgenden :

1. Bevölkerungszahl (in Millionen)
2. Fläche (in 1000 Quadratkilometer)
3. Bruttosozialprodukt (in Mrd. DM)
4. Anzahl der Erwerbstätigen (in Millionen)
5. Anteil der ausländischen Erwerbstätigen (in %)
6. Steueraufkommen (in Mrd. DM)
7. Schulden (in Mrd. DM)
8. Anteil der industriellen Produktion (in %)

Es versteht sich von selbst, daß bei Bedarf oder Interesse diese Liste verlängert werden könnte, oder daß an-

Voß C64/ Schule	Kapitel 8: Wirtschaft Abschnitt 6: Länder	Seite 283
-----------------------	--	--------------

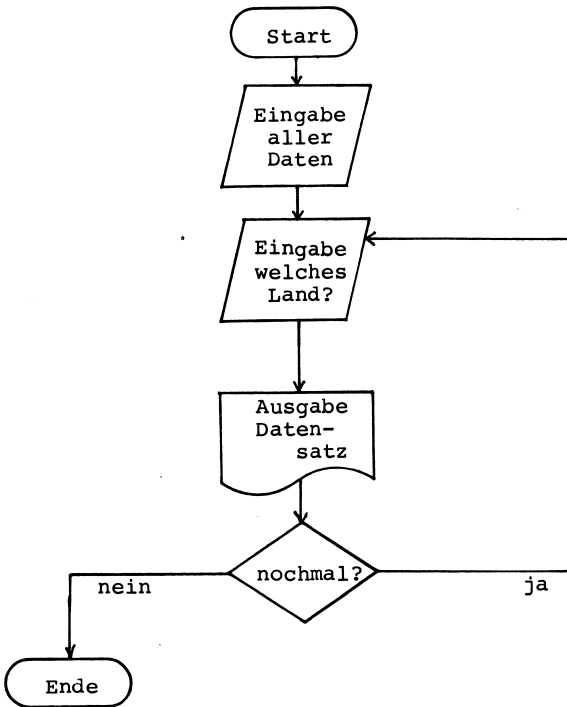
stelle der Bundesländer auch andere Sachverhalte (z.B. Nationen) betrachtet werden könnten. Natürlich müßte dann aber der Datenbestand entsprechend geändert werden.

②. SCHRITT : Problemanalyse

Der Leser wird schon festgestellt haben, daß sich über die verschiedenen Kapitel hinweg bestimmte Problemtypen wiederholen oder in ähnlicher Form wiederkehren. So ist dies auch bei dieser Problemstellung : Sie bietet im Prinzip keine neuen Probleme, so daß die Analyse recht einfach ist :

Es müssen zunächst alle relevanten Informationen eingegeben werden. Der Benutzer wird dann aufgefordert, anzugeben, über welches Land er Informationen wünscht. Nach der daraufhin erforderlichen Eingabe, erfolgt die Ausgabe des Datensatzes für das gewünschte Bundesland.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM W6-LAENDER
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUR AUSGABE VON INFORMATIONEN"
40 PRINT"UEBER DIE WIRTSCHAFTSKRAFT DER BUNDES-"
50 PRINTTAB(16)"LAENDER."
60 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT:PRINT
110 DIM L$(12),V(8,12),B$(8)
120 FOR I=1TO12:READ L$(I):NEXT I
130 FORI=1TO8:READB$(I):NEXTI
140 FOR J=1TO8:FORI=1TO12:READV(J,I):NEXTI:NEXTJ
150 GOSUB 1000:REM WARTEN
200 PRINT"FUER WELCHES LAND WERDEN INFORMATIONEN"
210 PRINT"GEWUENSCHT ? ":PRINT
220 FOR I=1 TO 12
230 PRINT"☐";TAB(5)"< ";I;" ";:PRINT"☐";TAB(13)L$(I)
240 NEXT I
250 PRINT:PRINT:INPUT"BITTE NUMMER ANGEBEN : ";Z
260 PRINTCHR$(147)
280 PRINT"☐";L$(Z):PRINT"☐":PRINT
290 FOR J=1TO8
300 PRINTB$(J);TAB(29)" : ";:PRINT"☐";V(J,Z);:PRINT"☐"
310 NEXT J
320 PRINT:PRINT:PRINT:GOSUB 1000:REM WARTEN
330 PRINT"NOCH EIN ANDERES LAND ? (J/N)":INPUT A$
340 IF A$="J" THEN PRINTCHR$(147):GOTO 200
350 PRINT:PRINT:PRINT"ENDE DER AUSGABE":END
500 DATA "SCHLESWIG-HOLSTEIN","NIEDERSACHSEN",
        "NORDRHEIN-WESTFALEN"
502 DATA "HESSEN","RHEINLAND-PFALZ","SAARLAND"
504 DATA "BADEN-WUERTEMBERG","BAYERN","HAMBURG","BREMEN",
        "BERLIN"
506 DATA "BUNDESREPUBLIK DEUTSCHLAND"
510 DATA BEVOELKERUNG(MILL.),FLAECHE(1000 GKM),
        SOZIALPRODUKT(MRD.DM)
512 DATA ERWERBSTAETIGE(MILL.),AUSLAENDERANTEIL(%),
        STEUERAUFKOMMEN MRD.DM)
514 DATA SCHULDEN(MRD.DM),ANTEIL INDUSTRIEPRODUKTION(%)
600 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
602 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
604 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
606 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 6 : Länder	Seite 286
-----------------------	--	--------------

```

608 DATA 0,0,0,0,0,0,0,0,0,0
610 DATA 0,0,0,0,0,0,0,0,0,0
612 DATA 0,0,0,0,0,0,0,0,0,0
614 DATA 0,0,0,0,0,0,0,0,0,0
616 DATA 0,0,0,0,0,0,0,0,0,0
618 DATA 0,0,0,0,0,0
1000 REM UP WARTEN
1010 PRINTTAB(5)"BITTE EINE TASTE DRUECKEN !"
1020 GET A$:IF A$="" THEN 1020
1030 PRINTCHR$(147)
1040 RETURN

```

5. SCHRITT : Variablenliste

A\$ = Stringvariable (Ja,Nein)
B\$ = Variablen-Bezeichnungen
I = Laufindex (Länder)
J = Laufindex (Variablen)
L\$ = Ländernamen
V = Variablen
Z = Nummer des gewünschten Landes

6. SCHRITT : Programmbeschreibung

Satz 10-70 : Überschrift

Satz 110 : Dimensionierungen

Satz 120-140 : Einlesen der Ausgangsdaten :

120 : Ländernamen

130 : Variablenbezeichnungen

140 : Variablenwerte

Satz 150 : Warten

Satz 200-250 : Anforderung der Nummer desjenigen Landes,

VoB C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 6 : Länder	Seite 287
<p style="text-align: center;">über das Informationen gewünscht werden, nachdem eine Länderliste auf dem Bildschirm vorgegeben wurde</p> <p>Satz 26φ-28φ : Löschen des Bildschirms und Ausgabe des Namens des gewünschten Landes</p> <p>Satz 29φ-31φ : Ausgabe des Datensatzes für das gewünschte Land</p> <p>Satz 32φ : Warten</p> <p>Satz 33φ-35φ : Abfrage, ob noch eine weitere Ausgabe gewünscht wird (wenn ja, Löschen des Bildschirms und zurück zu Satz 2φφ); wenn nein, Beendigung des Programms</p> <p>Satz 5φφ-5φ6 : Ländernamen in DATA-Statements.</p> <p>Satz 51φ-514 : Variablenbezeichnungen in DATA-Statements</p> <p>Satz 6φφ-618 : Variablenwerte :</p> <p style="text-align: center;">Die Werte sind so anzugeben, daß für die erste Variable alle Länderwerte, dann für die zweite Variable alle Länderwerte ... usw. folgen (vergl. Satz 14φ)</p> <p>ACHTUNG : IM PROGRAMM SIND NUR NULLEN (vergl. nächster Arbeitsschritt !)</p> <p>Satz 1φφφ-1φ4φ : Unterprogramm zum Warten.</p>		

Voß C64/ Schule	Kapitel 8 : Wirtschaft Abschnitt 6 : Länder	Seite 288
-----------------------	--	--------------

(7.) SCHRITT : Ergebnisse

Im Datenbestand des vorgestellten Programms finden sich nur Nullen, so daß bei einem Probelauf tatsächlich auch nur Nullwerte auf dem Bildschirm erscheinen können.

Wenn ein "echter" Probelauf durchgeführt werden soll, dann müssen erst konkrete Daten eingegeben werden, die der Leser der folgenden Übersicht entnehmen kann :

Wirtschaftsdaten der Bundesländer (die Daten beziehen sich auf 1980-1982) :

Land	Bevölk. (Mill.)	Fläche (1000 qkm)	Sozial prod. (Mrd.)	Erwerbs- tätige (Mill.)
Schleswig- Holstein	2.6	16	53	1.1
Niedersachsen	7.3	48	152	3.2
Nordrhein- Westfalen	17.0	34	406	6.9
Hessen	5.6	21	140	2.4
Rheinland-Pfalz	3.6	20	81	1.6
Baden-Württem- berg	9.3	36	234	4.3
Bayern	11.0	71	253	5.1
Saarland	1.1	3	24	.4
Hamburg	1.6	.7	75	.7
Bremen	0.7	.4	22	.3
Berlin	1.9	.5	53	.8
BUND	61.6	248	1484	26.8

Voß C64/ Schule	Kapitel 8: Wirtschaft Abschnitt 6: Länder	Seite 289		
Land	Ausländer- anteil	Steuer- aufkom- men (Mrd)	Schul- den (Mrd)	Anteil In- dustrie- produktion
Schleswig- Holstein	2.6	10	8	34
Niedersachsen	3.3	31	18	41
Nordrhein- Westfalen	7.1	108	32	48
Hessen	7.6	34	13	45
Rheinland-Pfalz	3.6	16	9	44
Baden-Württem- berg	10.1	57	18	52
Bayern	5.8	54	14	45
Saarland	5.7	4	3	51
Hamburg	8.4	33	8	32
Bremen	6.4	6	6	34
Berlin	10.7	12	7	35
BUND	6.7	365	233	45

Voß C64/ Schule	Kapitel 8: Wirtschaft Abschnitt 6: Länder	Seite 290

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 1 : Vorbemerkung	Seite 291
-----------------------	---	--------------



Kapitel 9 : Mathematik II
=====

9.1 Vorbemerkung

Abschließend soll ein Kapitel angefügt werden, in dem weitere mathematische Probleme, und zwar ausschließlich solche, die sich der sog. hochauflösenden Graphik bedienen, behandelt werden. Damit verfolgen wir im wesentlichen die nachstehenden Ziele :

1. Der schon etwas fortgeschrittenere Commodore-Benutzer soll erkennen können, daß mit relativ einfachen Programmen auch schon komplexere Problemstellungen in Angriff genommen werden können. Insoweit könnte dieses Schlußkapitel für den Anfänger eine leichte Überforderung darstellen (aber das gerade übt ja auch manchmal einen beträchtlichen Reiz aus).

2. Insbesondere aber soll gezeigt werden, wie bestimmte mathematische Probleme mit der hochauflösenden Graphik behandelt werden können, so daß der Leser auf diese Weise einen Zugang zur Lösung anderer geometrischer Probleme (und solcher aus dem Bereich der Analysis und der analytischen Geometrie) erfährt. Er wird dann erkennen, daß frühere Programme, in denen graphische Darstellungen versucht wurden, mit Hilfe der

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 1 : Vorbemerkung	Seite 292
<p>hochauflösenden Graphik sehr viel eleganter und anschaulicher angelegt werden können.</p> <p>Es ist klar, daß sich im Bereich der anspruchsvolleren mathematischen Fragestellungen unendlich viele Probleme anbieten, die per BASIC-Programm gelöst werden könnten. Es versteht sich deshalb von selbst, daß hier eine Auswahl getroffen werden muß, wobei wir gern zugeben, daß diese Auswahl relativ willkürlich ist. In einem Fortsetzungsband zu diesem Buch, der ausschließlich mathematischen Problemen gewidmet wird, werden wir diesbezüglich etwas systematischer vorgehen.</p>		

9.2 Graphik-Anweisungen

Wie schon an anderer Stelle erwähnt wurde, muß man sich bei der hochauflösenden Graphik den Bildschirm aufgeteilt vorstellen in 32ø Spalten und in 2øø Zeilen. So entstehen 64øøø einzelne Punkte, die im Prinzip einzeln ansprechbar sind.

Bei vielen BASIC-Rechnern stehen zur Programmierung der hochauflösenden Graphik, also zur Ansprache der Einzelpunkte, sehr hilfreiche Programmanweisungen zur Verfügung : So beispielsweise die Anweisung H PLOT (bei APPLE-Rechnern), mit deren Hilfe es möglich ist, Einzelpunkte oder auch ganze Linienzüge in hochauflösender Graphik auf den Bildschirm zu zeichnen.

Anweisungen mit ähnlichen Aufgabenstellungen sind bei anderen Rechnern zum Beispiel die Anweisungen PSET, DRAW, CIRCLE, PAINT und andere.

Der Commodore C 64 verfügt über derartige Anweisungen leider nicht. Deshalb gibt es nur zwei Wege, hoch aufgelöste Graphiken zu programmieren :

Entweder man programmiert tatsächlich Punkt für Punkt die Linienzüge, aus denen sich die gewünschte Graphik zusammensetzt (sicherlich ein recht aufwendiges Verfahren), oder man bedient sich leistungsfähiger Graphik-Hilfsprogramme, die es zu erschwinglichen Preisen zu kaufen gibt.

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 2 : Graphik-Anweisungen	Seite 294
-----------------------	--	--------------

Da wir in diesem Buch bisher davon ausgegangen sind, daß die vorgestellten Programme auch derjenige nutzbringend verwenden kann, der über keine zusätzliche Software und über keine Hilfs- und Dienstprogramme verfügt, soll der zweite Weg hier nicht weiter verfolgt werden.

Dies bedeutet, daß wir uns zunächst damit beschäftigen müssen, wie man auf möglichst geschickte Weise einzelne Bildschirmpunkte selbst programmiert. Wenn nämlich dieses Problem gelöst ist, dann kann man die erworbenen Kenntnisse dazu einsetzen, Linien zu programmieren (die sich ja aus einzelnen Punkten zusammensetzen), und damit ist man dann in der Lage, auch komplexere Graphiken zu erzeugen.

Bei der Darstellung der notwendigen Programmierschritte werden wir allerdings darauf verzichten, alle Einzelheiten im Detail zu erklären; vielmehr kommt es darauf an, den Weg in den Grundzügen zu skizzieren, der zur hochauflösenden Graphik führt :

Für die hochauflösende Graphik ist beim Commodore C 64 der sog. Videochip zuständig, dessen erste Speicheradresse den Wert 53248 hat. Diesen Wert wollen wir im folgenden, wie es in der Standardliteratur zum Commodore C 64 üblich geworden ist, der Variablen V zuweisen, also :

$$V = 53248$$

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 2 : Graphik-Anweisungen	Seite 295
-----------------------	--	--------------

Der erste Schritt, der bei der Erstellung einer hochauflösenden Graphik nun erforderlich ist, besteht in der "Umschaltung" des Rechners von der Textdarstellung in die Graphikdarstellung. Zu diesem Zweck muß in die Speicherstelle V+17 der Wert 59 geschrieben werden, also :

POKE V+17,59

Zusätzlich muß die Speicherstelle V+24 mit dem Wert 24 belegt werden, also

POKE V+24,24

Sind diese beiden Anweisungen abgeschickt, dann verschwindet der auf dem Bildschirm vorhandene Text (z.B. ein dort stehendes Programm) und wird ersetzt durch ein undefinierbares Graphikmuster.

Auch bei der hochauflösenden Graphik können die Farbmöglichkeiten des Commodore C 64 ausgenutzt werden, wobei jetzt (ökonomischerweise) die Farbsteuerung durch den aus der Normalgraphik schon bekannten Bildschirmspeicher übernommen wird (das waren, wie der Leser sich erinnern wird, die Bildschirmadressen 1ø24 bis 2ø23).

Die Angabe von Hintergrundfarbe (sie sei HF genannt) und Punktfarbe (PF) erfolgt gemeinsam in einer einzigen POKE-Anweisung, die sich auf Bildschirmspeicheradressen (1ø24 bis 2ø23) bezieht.

Die Ermittlung des in die Adresse per POKE einzuschrei-

VoB C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 2 : Graphik-Anweisungen	Seite 296
-----------------------	--	--------------

benden Wertes erfolgt gemäß dieser Anweisung :

POKE A, HF + 16*PF

Dabei ist A, die jeweilige Bildschirmspeicheradresse und HF und PF sind die Codezahlen für die beiden Farben.

Zur Erinnerung hier noch einmal die Farbtabelle :

Code	Farbe	Code	Farbe
0	schwarz	8	orange
1	weiß	9	braun
2	rot	10	hellrot
3	türkis	11	grau1
4	violett	12	grau2
5	grün	13	hellgrün
6	blau	14	hellblau
7	gelb	15	grau3

Will man also beispielsweise den ganzen Bildschirm mit einer Farbkombination aus Hintergrundfarbe und Punktfarbe belegen, will man z.B. weiß auf grün zeichnen, so hätte man zu programmieren :

```
FOR A = 1024 TO 2023
POKE A, 5+16*1
NEXT A
```

d.h. POKE A, 21

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 2 : Graphik-Anweisungen	Seite 297
-----------------------	--	--------------

Die hochauflösende Graphik selbst wird über die Speicheradressen 8192 bis 16383 gesteuert. Es ist sinnvoll, vor der Anfertigung einer hochauflösenden Graphik diesen Speicherbereich zu löschen, um frühere Graphik-"Reste" zum Verschwinden zu bringen.

Dieses Löschen geschieht durch Belegung dieser Speicheradressen mit null, also :

```
FOR A = 8192 TO 16383
POKE A,0
NEXT A
```

Wenn diese Anweisungsfolge ausgeführt wird, verschwinden alle Graphikreste auf dem Bildschirm, was eine Weile dauert und in Ruhe beobachtet werden kann.

Ohne weitere Erklärung soll nun ein kleines Programm vorgestellt werden, welches dazu taugt, an eine bestimmte Stelle des "hochaufgelösten" Bildschirms einen Punkt zu zeichnen. Wenn wir dabei auf eine Farbgebung, wie sie oben erläutert wurde, verzichten, wird per Voreinstellung schwarz auf hellblau gezeichnet.

Die Lage des zu zeichnenden Punktes wird durch zwei Koordinatenwerte charakterisiert, nämlich den X-Wert, der die Bildschirmspalte angibt (\emptyset bis 319) und dem Y-Wert, der die Bildschirmzeile angibt (\emptyset -199).

Diese beiden Werte werden in dem folgenden Programm so in Werte XK und YK transformiert, daß aus ihnen die Speicheradresse S berechnet werden kann (Satz $2\emptyset4\emptyset$), in die dann der Punkt per POKE gezeichnet

werden kann (dies geschieht in Satz 2050).

```
2000 REM UP PUNKT
2010 XK=8*INT(X/8)
2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX=2*(7-INT((X/8-INT(X/8))*8))
2040 S=8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN
```

Dieses Programm dient also dazu, daß der Rechner sich selbst ausrechnet, welcher Speicher wie beschrieben werden soll, wenn an einer ganz bestimmten Stelle des Bildschirms ein Punkt gezeichnet werden soll.

Dies bedeutet, daß dieses kleine Programm als Unterprogramm in jedes Programm eingebaut werden kann, das eine hochauflösende Graphik erstellen soll, die sich ja aus einzelnen Punkten zusammensetzt. Bei der Erstellung einer Graphik braucht dieses Unterprogramm dann nur mehrfach angesprungen zu werden.

In einigen der folgenden Abschnitte wird dies an einfachen Beispielen anschaulich demonstriert.

ANMERKUNG : Wenn in einem Graphik-Programm ein Fehler passiert, dann stoppt das Programm im Graphikmodus, d.h. ohne lesbare Fehlermeldung. Aus dem Graphikmodus gelangt man zurück in den Textmodus durch Drücken der Tasten RUN/STOP und RESTORE gleichzeitig.

9.3 Die Gerade

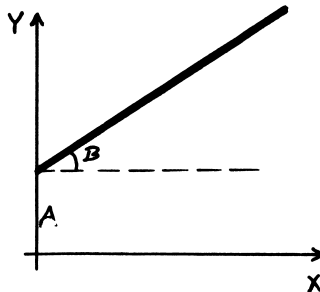
① SCHRITT : Vorstellung des Problems

In einem ersten Beispiel zur Verwendung der hochauflösenden Graphik soll ein Programm vorgestellt werden, welches eine Gerade zeichnet. Dabei sollen die Parameter dieser Geraden (Ordinatenabschnitt und Steigung) alternativ besetzbar sein.

② SCHRITT : Problemanalyse

Eine Gerade kann im Achsenkreuz gemäß der nebenstehenden Skizze durch zwei Parameterwerte fixiert werden :

1. Durch den Ordinatenabschnitt, der mit A bezeichnet wird;
2. durch die Steigung (Tangens des Steigungswinkels), die mit B bezeichnet wird.



Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 3 : Gerade	Seite 300
-----------------------	---	--------------

In dem zu entwickelnden Programm müssen also zunächst die beiden Parameter vorgegeben werden.

In einem zweiten Arbeitsschritt ist dann der Bildschirm für die hochauflösende Graphik vorzubereiten, wonach dann als erstes ein Achsenkreuz gezeichnet werden kann.

Bei der Zeichnung der Geraden ist von der aus der Geometrie bekannten Funktionalgleichung auszugehen, welche lautet :

$$y_i = a + b \cdot x_i$$

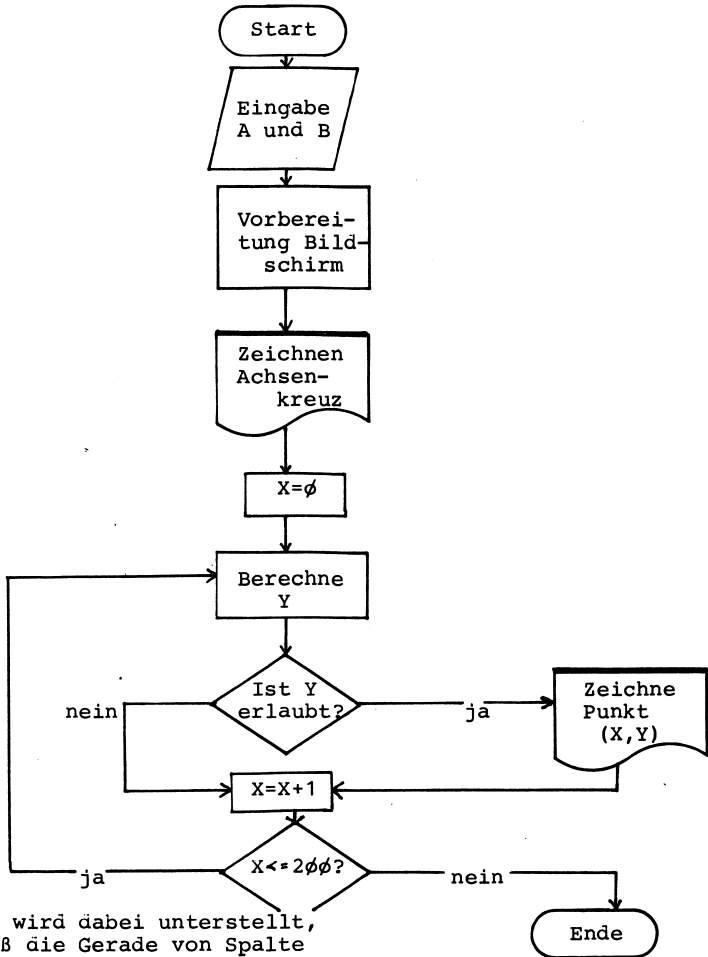
bzw. in BASIC-Schreibweise :

$$Y = A + B \cdot X$$

Es ist dabei zu beachten, daß die Bildschirmzeilen von oben nach unten numeriert sind (Ø bis 199). Wenn wir die aus der Funktionalgleichung sich ergebenden Y-Werte direkt als Bildschirmzeilenwerte interpretieren würden, müßten höhere Y-Werte immer tiefer auf dem Bildschirm sein.

Dies aber entspricht nicht der geometrischen Vorstellung, nach der höhere Y-Werte höher im Koordinatensystem plazierte sind als niedrigere. Wir müssen deshalb eine geeignete subtraktive Umrechnung vornehmen.

3. SCHRITT : Flußdiagramm



Es wird dabei unterstellt, daß die Gerade von Spalte 0 bis Spalte 200 gezeichnet wird, also nicht über die gesamte Bildschirmbreite.

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 3 : Gerade	Seite 302
-----------------------	---	--------------

④ 4. SCHRITT : Programm

```

10 REM R1-GERADE
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUM ZEICHNEN EINER GERADEN IN"
40 PRINT"HOCHAUFLOESENDER GRAPHISCHEN DARSTELLUNG"
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT:PRINT
70 PRINT"DIESES PROGRAMM BENÖTIGT ALS EINGABE-"
80 PRINT"INFORMATIONEN DIE STEIGUNG B UND DEN "
90 PRINT"ORDINATENABSCHNITT A DER GERADEN."
100 PRINT:INPUT"STEIGUNG      : ";B
110 PRINT:INPUT"ACHSENABSCHNITT : ";A
120 REM VORBEREITUNG DES BILDSCHIRMS
130 GOSUB 1900:REM SCREEN
140 REM ACHSENKREUZ
150 FOR X=0 TO 200
160 Y=130
170 GOSUB 2000:REM PUNKT
180 NEXT X
190 FOR Y=20 TO 150
200 X=0
210 GOSUB 2000:REM PUNKT
220 NEXT Y
230 REM GERADE
240 FOR X=0 TO 200
250 Y=130-B*X-A
255 IF Y<0 OR Y>199 THEN 270
260 GOSUB 2000:REM PUNKT
270 NEXT X
280 REM WARTEN
290 FOR I=1 TO 2000:NEXT I
300 REM TEXT
310 GOSUB 2100:REM TEXT
320 PRINTCHR$(147)
399 PRINT"ENDE":END
1899 REM HGR-HILFEN
1900 REM UP SCREEN
1910 V=53248
1920 POKE V+17,59:POKE V+24,24
1930 FOR I=1024 TO 2023:POKE I,14:NEXT I
1940 FOR I=8192 TO 16383:POKE I,0:NEXT I
1950 RETURN

```

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 3 : Gerade	Seite 303
-----------------------	---	--------------

```

2000 REM UP PUNKT
2010 XK=8*INT(X/8)
2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX=2*(7-INT((X/8-INT(X/8))*8))
2040 S=8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN
2100 REM UP TEXT
2110 POKEV+17,155:POKEV+24,21
2120 RETURN

```

5. SCHRITT : Variablenliste

- A = Ordinatenabschnitt der Geraden
- B = Steigung der Geraden
- EX = Hilfsgröße beim Zeichnen
- I = Laufindex
- S = Speicheradresse beim Zeichnen eines Punktes
- V = Startadresse des Videochips
- X = X-Koordinatenwert
- XK = Transformierter X-Wert zum Zwecke des Zeichnens eines Punktes
- Y = Y-Koordinatenwert
- YK = Transformierter Y-Wert

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 3 : Gerade	Seite 304
-----------------------	---	--------------

6. SCHRITT : Programmbeschreibung

Satz 1 ϕ -9 ϕ : Überschrift und Erläuterungen

Satz 1 $\phi\phi$ -11 ϕ : Eingabe der Input-Informationen (Ordinatenabschnitt und Steigung)

Satz 12 ϕ -13 ϕ : Sprung ins Unterprogramm 19 $\phi\phi$: Vorbereitung des Bildschirms

Satz 14 ϕ -18 ϕ : Zeichnen der waagrechten Achse von Spalte ϕ bis 2 $\phi\phi$ in Zeile 13 ϕ (willkürlich ausgewählt)

diese Achse setzt sich aus 2 ϕ 1 Punkten zusammen, weshalb in der X-Schleife 2 ϕ 1 mal das Unterprogramm 2 $\phi\phi\phi$ angesprungen wird (Zeichnen eines Punktes)

Satz 19 ϕ -22 ϕ : Zeichnen der senkrechten Achse in Spalte ϕ von Zeile 2 ϕ bis 15 ϕ (Details wie oben)

Satz 23 ϕ -27 ϕ : Zeichnen der Geraden zwischen der ϕ . und der 2 $\phi\phi$. Spalte

Wegen der Art der Bildschirmzeilenzählung muß dabei die Geradengleichung transformiert werden in :

$$Y = 13\phi - B * X - A$$

(die Zahl 13 ϕ entspricht dabei der Zeilennummer der waagrechten Achse)

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 3 : Gerade	Seite 305
<p>Satz 28ø-29ø : Einbau einer Warteschleife, damit der Benutzer die Graphik studieren kann</p> <p>Satz 3øø-31ø : Umschalten in den Textmodus durch Sprung in das Unterprogramm 21øø</p> <p>Satz 32ø-99ø : Beendigung des Programms nach Löschung des Bildschirms</p> <p>Satz 1899-195ø : Unterprogramm Löschen des Bildschirms:</p> <p style="padding-left: 40px;">191ø : Vorgabe der Startadresse</p> <p style="padding-left: 40px;">192ø : Umschalten in den Graphikmodus</p> <p style="padding-left: 40px;">193ø : Löschen des Bildschirms im Graphikmodus</p> <p style="padding-left: 40px;">194ø : Löschen des Speichers für hochauflösende Graphik</p> <p style="padding-left: 40px;">195ø : Rücksprung</p> <p>Satz 2øøø-2ø6ø : Unterprogramm Punkt</p> <p style="padding-left: 40px;">Dieser Programmteil wurde in Abschnitt 9.2 besprochen</p> <p>Satz 21øø-212ø : Umschalten in den Textmodus</p>		

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 3 : Gerade	Seite 306
-----------------------	---	--------------

⑦. SCHRITT : Ergebnisse

Die Ergebnisse dieses Programms brauchen nicht vorgestellt zu werden. Sie entsprechen - in Abhängigkeit von den beiden eingegebenen Parametern A und B - der Skizze, die wir im ersten Arbeitsschritt vorgestellt haben.

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 4 : Kreis	Seite 307
-----------------------	--	--------------

9.4 Der Kreis

①. SCHRITT : Vorstellung des Problems

Nach dem gleichen Muster wie im vorangegangenen Abschnitt ein Programm zum Zeichnen einer Geraden entwickelt wurde, soll nun ein Programm zum Zeichnen beliebiger Kreise vorgestellt werden. Dabei wiederholt sich nun vieles, so daß wir uns entsprechend kurz fassen können.

②. SCHRITT : Problemanalyse

Wir stehen hier vor dem gleichen Problem wie im Abschnitt zuvor, mit dem Unterschied, daß wir nun von der Kreisgleichung ausgehen müssen :

Diese Kreisgleichung lautet allgemein :

$$(x - a)^2 + (y - b)^2 = r^2$$

Dabei ist r der Radius des Kreises, x und y sind die Koordinaten der Kreislinie und a und b sind die Koordinaten des Kreismittelpunktes.

Löst man diese Beziehung nach y auf (Lösen einer quadratischen Gleichung), so erhält man die Gleichungen für den oberen und den unteren Halbkreis.

Es gilt nämlich :

$$(y - b)^2 = r^2 - (x - a)^2 = D$$

$$y - b = \pm \text{SQR}(D)$$

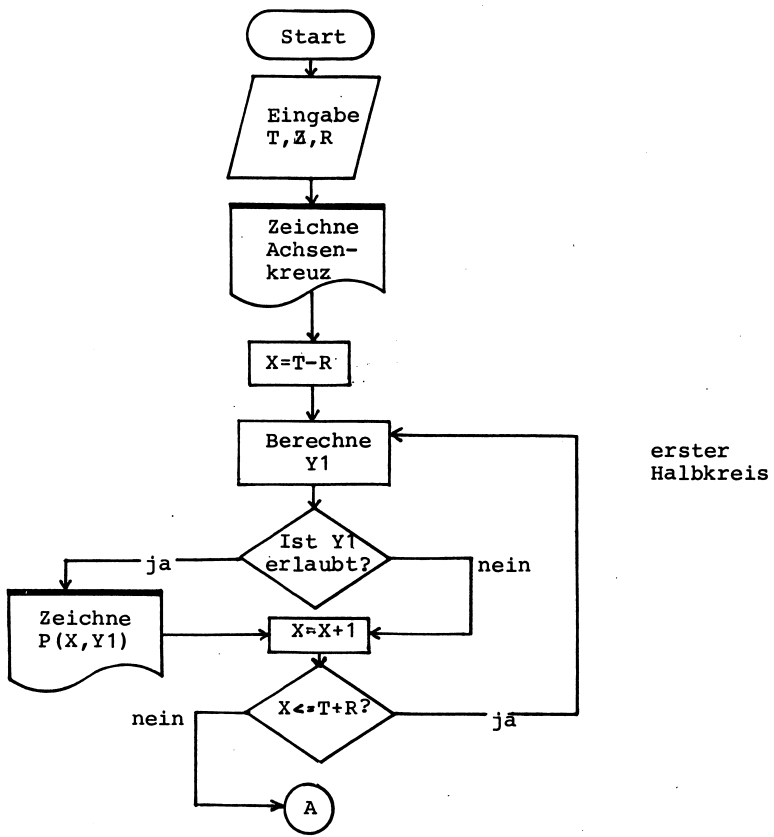
$$Y_{1/2} = b \pm \text{SQR}(D)$$

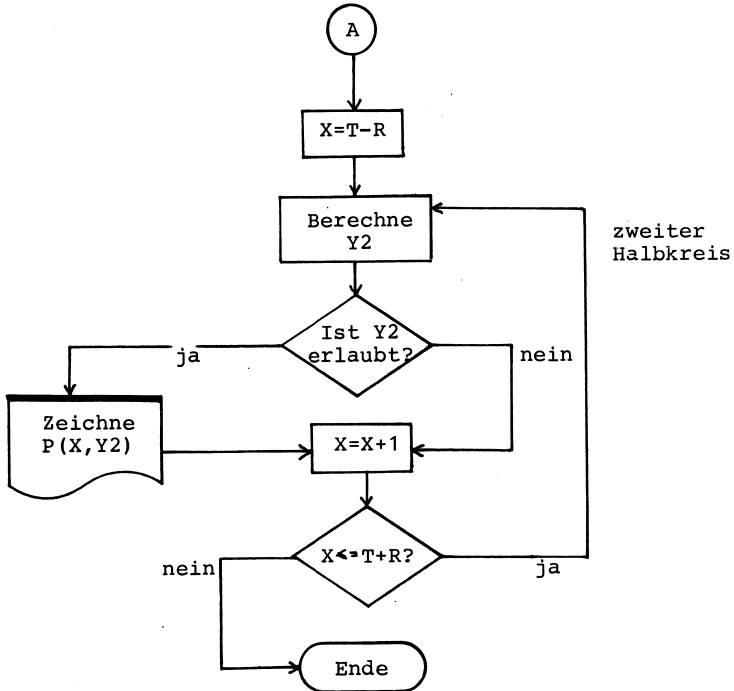
Damit sind die Grundlagen für das folgende Programm klar :

Es benötigt als Input-Informationen die Koordinaten des Kreismittelpunktes a und b (wir nennen sie im folgenden Programm T und Z) und den Radius r (in BASIC : R). Dann können für alternative Werte von X die dazugehörigen Werte $Y1$ und $Y2$ gemäß der obigen Beziehung bestimmt werden und die Punkte $X,Y1$ und $X,Y2$ können gezeichnet werden.

Es ist dabei zu beachten, daß sich X nur sinnvollerweise im Bereich von $T-R$ bis $T+R$ bewegen kann, weil außerhalb dieses Bereichs keine Y -Werte im reellen Bereich definiert sind (die Hilfsgröße D würde dort negativ und der Versuch, aus ihr die Wurzel zu ziehen, würde zu einer Fehlermeldung und zum Programmabbruch führen).

3. SCHRITT : Flußdiagramm





Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 4 : Kreis	Seite 311
<p style="text-align: center;">4. SCHRITT : Programm</p> <pre> 10 REM R2-KREIS 20 PRINTCHR\$(147) 30 PRINT"PROGRAMM ZUM ZEICHNEN EINES BELIEBIGEN" 40 PRINTTAB(16)"KREISES." 50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984" 60 PRINT:PRINT:PRINT:PRINT 70 PRINT"DIESES PROGRAMM BENÖTIGT ALS INPUT-" 80 PRINT"INFORMATIONEN : ":PRINT : ";T 110 PRINTTAB(5);:INPUT"Y-WERT (ZEILE) : ";Z 120 PRINT:INPUT"RADIUS : ";R 130 PRINTCHR\$(147) 140 REM BILDSCHIRM 150 GOSUB 1900:REM SCREEN 160 REM ACHSENKREUZ 1 170 FORK=0TO319 180 Y=199:GOSUB 2000:REM PUNKT 190 NEXT X 200 FOR Y=0 TO199 210 X=1:GOSUB2000:REM PUNKT 220 NEXT Y 240 REM ACHSENKREUZ 2 250 B=T+R+10 270 IF B>319 THEN B=319 280 FOR X=0TOB 290 Y=199-Z 300 GOSUB 2000:REM PUNKT 310 NEXT X 320 A=199-Z-R-10 330 IF A<0THEN A=0 350 FOR Y=A TO 199 360 X=T 370 GOSUB 2000:REM PUNKT 380 NEXT Y 390 REM KREIS 400 A=T-R:B=T+R 410 IF A<0 THEN A=0 420 IF B>319 THEN B=319 430 FOR X=A TO B 435 D=R*R-(X-T)^2 437 IF D<=0 THEN Y=199-Z:GOTO 460 440 Y=199-Z+SQR(D) 450 IF Y<0 OR Y>199 THEN 470 460 GOSUB 2000:REM PUNKT 470 NEXT X </pre>		

VoB C64/ Schule	Kapitel 9: Mathematik II Abschnitt 4: Kreis	Seite 312
<pre> 480 FOR X=BTOA STEP-1 485 D=R*R-(X-T)^2 487 IF D<=0 THEN Y=199-Z:GOTO 510 490 Y=199-Z-SQR(D) 500 IF Y<0 OR Y>199 THEN 520 510 GOSUB 2000:REM PUNKT 520 NEXT X 530 REM WARTEN 540 FOR I=1TO5000:NEXT I 550 REM TEXT 560 GOSUB 2100:REM TEXT 570 PRINTCHR\$(147) 999 PRINT"ENDE":END 1899 REM HGR-HILFEN 1900 REM UP SCREEN 1910 V=53248 1920 POKE V+17,59:POKE V+24,24 1930 FOR I=1024TO2023:POKE I,14:NEXT I 1940 FOR I=8192TO16383:POKE I,0:NEXT I 1950 RETURN 2000 REM UP PUNKT 2010 XK=8*INT(X/8) 2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8) 2030 EX=2*(7-INT((X/8-INT(X/8))*8)) 2040 S=8192+XK+YK 2050 POKE S,PEEK(S) OR EX 2060 RETURN 2100 REM UP TEXT 2110 POKEV+17,155:POKEV+24,21 2120 RETURN </pre>		

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 4 : Kreis	Seite 313
<p data-bbox="347 331 779 379">5. SCHRITT : Variablenliste</p> <p data-bbox="208 453 962 539">A = Senkrechte Koordinatenwerte für das zweite Achsenkreuz, das hilfweise im Kreismittelpunkt errichtet wird</p> <p data-bbox="303 555 936 608">später Anfangswert der X-Koordinatenwerte des zu zeichnenden Kreises</p> <p data-bbox="208 624 962 676">B = Waagrechte Koordinatenwerte für das zweite Achsenkreuz</p> <p data-bbox="303 692 947 715">später Endwert der X-Koordinatenwerte des Kreises</p> <p data-bbox="208 724 725 746">D = Hilfsgröße in der Kreisgleichung</p> <p data-bbox="208 756 801 778">EX = Hilfsgröße beim Zeichnen eines Punktes</p> <p data-bbox="208 788 419 810">I = Laufindex</p> <p data-bbox="208 820 448 842">R = Kreisradius</p> <p data-bbox="208 852 740 874">S = Speicherstelle beim Punktzeichnen</p> <p data-bbox="208 884 685 906">T = Spalte des Kreismittelpunktes</p> <p data-bbox="208 916 648 938">V = Startadresse des Videochip</p> <p data-bbox="208 948 827 970">X = waagrechte Koordinate der Kreisgleichung</p> <p data-bbox="208 979 659 1002">XK = transformierte X-Koordinate</p> <p data-bbox="208 1011 827 1034">Y = senkrechte Koordinate der Kreisgleichung</p> <p data-bbox="208 1043 659 1066">YK = transformierte Y-Koordinate</p> <p data-bbox="208 1075 670 1098">Z = Zeile des Kreismittelpunktes</p>		

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 4 : Kreis	Seite 314
-----------------------	--	--------------

6. SCHRITT : Programmbeschreibung.

Satz 10-90 : Überschrift und Erläuterungen

Satz 100-120 : Vorgabe der Ausgangsinformationen

Satz 130-150 : Vorbereitung des Bildschirms für die hochauflösende Graphik.

Satz 160-190 : Zeichnen der waagrechten Achse des Hauptachsenkreuzes (erstes Achsenkreuz)

Satz 200-220 : Zeichnen der senkrechten Achse des Hauptachsenkreuzes

Satz 240-380 : Zeichnen eines zweiten Achsenkreuzes (Hilfsachsenkreuz) durch den Mittelpunkt des zu zeichnenden Kreises.

Dieses Hilfsachsenkreuz wird so gezeichnet, daß die Hilfsachsen bei den Hauptachsen (s.o.) beginnen und die Kreislinie in entgegengesetzter Richtung um 10 Spalten bzw. 10 Zeilen überqueren (vergl. Skizze im 7. Arbeitsschritt)

Satz 390-470 : Zeichnen des ersten Halbkreises :

400 : Anfangs- und Endwert (A und B) der Bewegung der Variablen X

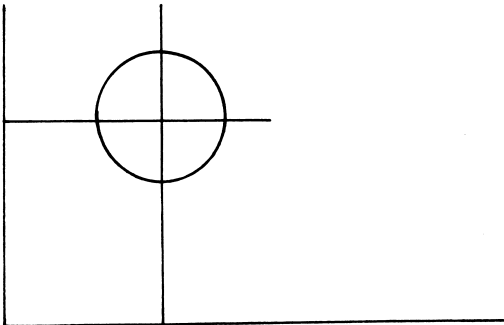
410 : Kontrolle von A ...

420 : ... und Kontrolle von B auf Zulässigkeit

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 4 : Kreis	Seite 315
<p style="text-align: center;">43φ : Beginn der X-Schleife (d.h. Beginn der Bestimmung der Kreislinienkoordinaten)</p> <p style="text-align: center;">435 : Bestimmung der Hilfsgrößen D</p> <p style="text-align: center;">437 : Kontrolle dieser Größe auf Zulässigkeit</p> <p style="text-align: center;">44φ : Bestimmung der Y-Koordinate</p> <p style="text-align: center;">45φ : Kontrolle von Y auf Zulässigkeit</p> <p style="text-align: center;">46φ : Zeichnen von P(X,Y)</p> <p style="text-align: center;">47φ : Nächster X-Wert</p> <p>Satz 48φ-52φ : Gleiche Prozedur wie in 39φ-47φ nun für den zweiten Halbkreis (Details s.o.)</p> <p>Satz 53φ-54φ : Warteschleife</p> <p>Satz 55φ-999 : Umschalten in den Textmodus, Löschen des Bildschirms und Beendigung des Programms</p> <p>Satz 1899-194φ: Vorbereiten des Bildschirms für die hochauflösende Graphik (siehe 9.3)</p> <p>Satz 2φφφ-2φ6φ: Zeichnen eines Punktes (siehe 9.2)</p> <p>Satz 21φφ-212φ: Umschalten in den Textmodus (siehe 9.2)</p>		

7. SCHRITT : Ergebnisse

In Abhängigkeit von den eingegebenen Parameterwerten
(Mittelpunktkoordinaten und Radius des zu zeichnenden
Kreises) könnte das folgende Bild entstehen :



Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 5 : Sinus	Seite 317
-----------------------	--	--------------

9.5 Die Sinuslinie

①. SCHRITT : Vorstellung des Problems

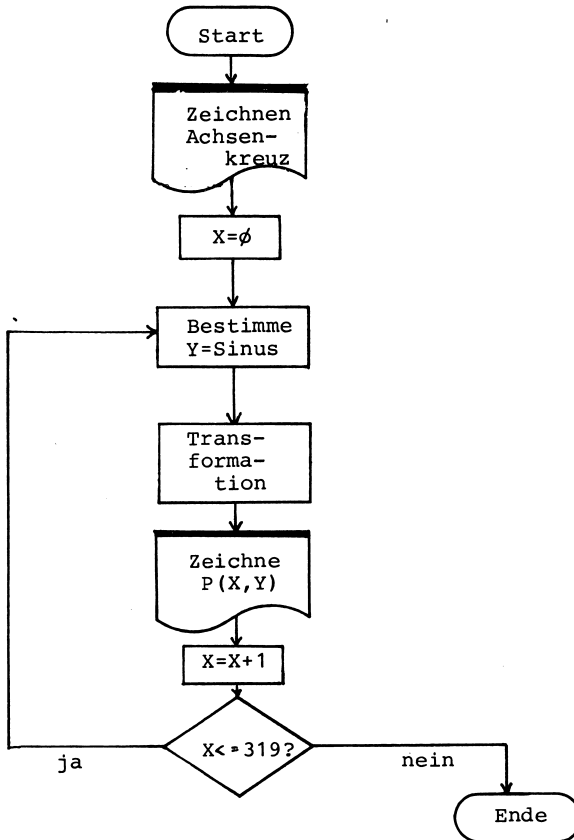
In einem letzten Beispiel hochauflösender Graphik wollen wir eine Sinusschwingung darstellen, der wir im Kapitel "Physik" schon einmal begegnet waren. Die Vorgehensweise ist die gleiche wie in den vorangegangenen beiden Abschnitten.

②. SCHRITT : Problemanalyse

Die Problemanalyse ist hier einfach, weil keine neuen Probleme zu bewältigen sind. Wir haben lediglich darauf zu achten, daß die Periodizität der Sinusschwingung (Frequenz) und ihre Amplitude (Ausschlag) so eingerichtet werden, daß die auf dem Bildschirm auszugebende Schwingung optisch gut aussieht, d.h. den zur Verfügung stehenden Raum in geschickter Weise ausnutzt.

3. SCHRITT : Flußdiagramm

Den vorstehenden Überlegungen entspricht es, daß wir uns beim Flußdiagramm auch sehr kurz fassen können :



VoB C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 5 : Sinus	Seite 319
-----------------------	--	--------------

4. SCHRITT : Programm

```

10 REM R3-SINUS
20 PRINTCHR$(147)
30 PRINT"PROGRAMM ZUM ZEICHNEN EINER SINUSLINIE"
50 PRINT:PRINT:PRINTTAB(9)"PROF.DR.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINTCHR$(147)
80 GOSUB 1900:REM SCREEN
90 REM ACHSENKREUZ
100 FOR X=0 TO 319
110 Y=100
120 GOSUB 2000:REM PUNKT
130 NEXTX
140 FOR X=0 TO 319 STEP 39.5
150 Y=101
160 GOSUB 2000:REM PUNKT
170 NEXT X
180 FORY=60 TO 140
190 X=2
200 GOSUB 2000:REM PUNKT
210 NEXT Y
220 FOR Y=60 TO 140 STEP 10
230 X=1
240 GOSUB 2000:REM PUNKT
250 NEXT Y
280 REM SINUS
290 FOR X=0TO319
300 P=120:F=40
310 Y=SINK X/25)
320 Y=F*Y:Y=100-Y
330 GOSUB 2000:REM PUNKT
340 NEXT X
350 REM WARTEN
360 FOR I=1TO5000:NEXTI
370 GOSUB 2100:REM TEXT
380 PRINTCHR$(147)
390 PRINT"ENDE DER AUSGABE":END
1899 REM HGR-HILFEN
1900 REM UP SCREEN
1910 V=53248
1920 POKE V+17,59:POKE V+24,24
1930 FORI=1024TO2023:POKEI,14:NEXTI
1940 FORI=8192TO16383:POKEI,0:NEXTI
1950 RETURN

```

```
2000 REM UP PUNKT
2010 XK=8*INT(X/8)
2020 YK=320*INT(Y/8)+INT((Y/8-INT(Y/8))*8)
2030 EX=2+(7-INT((X/8-INT(X/8))*8))
2040 S=8192+XK+YK
2050 POKE S,PEEK(S) OR EX
2060 RETURN
2100 REM UP TEXT
2110 POKEV+17,155:POKEV+24,21
2120 RETURN
```

5. SCHRITT : Variablenliste

- EX = Hilfsgröße beim Zeichnen eines Punktes
- F = Transformationsfaktor bei der Amplitudenbestimmung.
- I = Laufindex
- P = Transformationsfaktor für die Periodizität der Schwingung.
- S = Speicherstelle beim Punktzeichnen
- V = Startadresse Videochip
- X = Spaltenkoordinatenwert
- XK = Transformierter X-Wert beim Punktzeichnen
- Y = Zeilenkoordinatenwert
- YK = Transformierter Y-Wert beim Punktzeichnen

6. SCHRITT : Programmbeschreibung

Satz 1φ-6φ : Überschrift

Satz 7φ-8φ : Vorbereitung des Bildschirms für die
hochauflösende Graphik (per Unterpro-
gramm 19φφ)

Satz 1φφ-25φ : Zeichnen eines Achsenkreuzes :

1φφ-13φ : waagrechte Achse

14φ-17φ : Markierungen an die waag-
rechte Achse

18φ-21φ : senkrechte Achse

22φ-25φ : Markierungen an die senk-
rechte Achse

Satz 28φ-34φ : Zeichnen der Sinuslinie :

29φ : Vorgabe der Bildschirmspalten

30φ : Vorgabe der Skalenparameter

31φ : Bestimmung des Sinuswertes

32φ : Bestimmung des transformier-
ten Sinuswertes

33φ : Zeichnen per Unterprogramm
2φφφ

Satz 35φ-36φ : Warteschleife

Satz 37φ : Umschalten in den Textmodus per Unter-
programm 21φφ

Voß C64/ Schule	Kapitel 9 : Mathematik II Abschnitt 5 : Sinus	Seite 322
-----------------------	--	--------------

Satz 38φ-39φ : Löschen des Bildschirms und Beendi-
gung des Programms

Satz 1899-195φ : Vorbereitung des Bildschirms für die
hochauflösende Graphik.

Satz 2φφφ-2φ6φ : Unterprogramm zum Zeichnen eines
Punktes in hochauflösender Graphik

Satz 21φφ-2φ2φ : Unterprogramm zum Umschalten in den
Textmodus

⑦ SCHRITT : Ergebnisse

Die Ergebnisse dieses Programms brauchen hier nicht
vorgestellt zu werden. Es entsteht eine Sinuslinie
in der bekannten Gestalt.

Voß C64/ Schule	Kapitel 10 : Ausblick Abschnitt - : -	Seite 323
<p>Ausblick =====</p> <p>In den vorangegangenen Kapiteln haben wir versucht, zu zeigen, wie mit Hilfe vergleichsweise einfacher BASIC-Programme typische Probleme, wie sie im Schulalltag auftreten, gelöst werden können.</p> <p>Mit Nachdruck muß darauf hingewiesen werden, daß die ausgewählten Beispiele nur exemplarischen Charakter haben, und daß wir keinen Wert darauf gelegt haben, optimale Programme zu erstellen.</p> <p>Der Leser sollte vielmehr in die Lage versetzt werden, die Funktionsweise der Programme rasch nachvollziehen zu können. Damit wird er dann in der Lage sein, auch für andere Probleme, die ja häufig den hier vorgestellten Beispielen ähneln werden, die entsprechenden Lösungsansätze zu finden.</p> <p>Entscheidend beim Computereinsatz ist nämlich nicht der Rechner selbst, sondern die Fähigkeit des Benutzers, bestimmte Probleme so aufzubereiten, daß er dem Rechner den adäquaten Lösungsweg in Form eines Programms geben kann.</p> <p>Wichtig ist nämlich nicht in erster Linie, daß ein Programm schon beim ersten Versuch fehlerfrei läuft (auch wenn dies natürlich sehr zu begrüßen und erfreulich ist), sondern daß der Computerbenutzer es lernt, wie ein gestelltes Problem gedanklich vorstrukturiert werden muß, damit es durch einen Rechner dann bearbeitet werden kann :</p>		

Voß C64/ Schule	Kapitel 10 : Ausblick Abschnitt - : -	Seite 324
-----------------------	--	--------------

Ein Computer nützt uns nämlich nichts, wenn wir, die Benutzer, die jeweiligen Problemlösungswege nicht schon im Kopf haben.

Und wenn einmal ein Programm nicht so funktionieren sollte, wie der Benutzer das erwartet, und zum Beispiel mit einer Fehlermeldung der Programmablauf abgebrochen wird, dann hilft immer noch ein Blick in das Benutzerhandbuch des Commodore C 64, wo sich im Anhang L Fehlermeldungen und Erklärungen dazu finden.

Für diejenigen Leser, denen die vorgestellten Beispiele zu einfach waren, sei darauf hingewiesen, daß dieses Buch in erster Linie für BASIC-Einsteiger gedacht war, die wir nicht schon durch die eventuelle Komplexität der Aufgabenstellungen überfordern wollten. Zu ihrem Trost aber verweisen wir auf einen in Bälde erscheinenden Fortsetzungsband zu diesem Buch, in dem dann insbesondere komplexere und vor allem mathematische Fragestellungen aufgegriffen werden sollen.

Voß C64/ Schule	Stichwortverzeichnis	Seite 325
** A **		
	ABBILDUNG, OPTISCHE	163
	ABFRAGE	28
	ABS	25
	ADDITION	23
	ADRESSE	140
	ANFANGSWERT	32
	ARBEITSSCHRITT	41
	ARCHIMEDES	151
	ARGUMENT	24
	ARRAY	11
	AUSGABE	12
	AUSPRAEGUNG	10
** B **		
	BASIC	7, 17
	BASIC-ANWEISUNG	95, 179
	BASIC-GRAPHIKANWEISUNG	136
	BETRIEBSSYSTEM	12
	BEVOELKERUNGSENTWICKLUNG	245
	BILDSCHIRM	13, 132
	BILDSCHIRMADRESSE	133
	BILDSCHIRMSPALTE	132
	BILDSCHIRMZEILE	132
	BILDSCHIRMZONE	21
	BIOLOGIE	211
	BIT	9
	BRUTTOBETRAG	80
	BUCHSTABE	9
	BUNDESLAENDER	282
	BYTE	9
** C **		
	CHEMIE	95
	CHR\$	25
	CONT	36
	COS	25
	CTRL	97

Voß C64/ Schule	Stichwortverzeichnis	Seite 326
	** D **	
	DATA	99
	DATEI	11
	DATEN	8, 12
	DATENEINGABE	99
	DATENSATZ	11
	DATENVERARBEITUNG	8
	DIM	100
	DIMENSIONIERUNG	100
	DISKETTE	13, 35
	DIVISION	23
	DREIECK, RECHTWINKLIGES	42
	DREISATZ	81
	DREISATZRECHNUNG	80
	DRUCKER	13
	** E **	
	E	74
	EINGABE	12
	EINGABEFEHLER	40
	ELEMENT, CHEMISCHES	122
	END	18
	ENDLOSSCHLEIFE	36
	ENDWERT	32
	ENGLISCH-VOKABELN	196
	ENGLISCH-VOKABELTEST	196
	ERDKUNDE	231
	ERGEBNIS	41
	ERGEBNISAUSGABE	17
	EVA-PRINZIP	42, 152
	EXP	25
	EXPONENTIALDARSTELLUNG	92
	** F **	
	FARBADRESSE	141
	FARBE	97, 98, 132, 295, 296
	FARBSPEICHERADRESSE	134
	FEDERWAAGE	144
	FELD	10
	FILE	11
	FLUSSDIAGRAMM	14, 41
	FOR...TO	32
	FRANZOESISCH-VOKABELN	189
	FREMDSPRACHE	179
	FUNKTION	24, 25
	FUNKTIONSNAME	24
	FUNKTIONSWEISE	12

Voß C64/ Schule	Stichwortverzeichnis	Seite 327
	** G **	
	GERADE	299
	GESCHICHTE	231
	GESETZ, OHMSCHES	173
	GET	180
	GGT	50
	GLEICHUNG, QUADRATISCHE	68
	GOSUB	180
	GOTO	30
	GRAPHIK	132, 293
	GRAPHIK, HOCHAUFLOESENDE	132, 291
	GRAPHIK-ANWEISUNG	293
	GRAPHIK-PROGRAMMIERUNG	132
	GRAPHIKZEICHEN	96
	** H **	
	HAEUFIGKEITSVERTEILUNG	273
	HAUPTSTAEDTE	238
	HYPOTHEK	262
	HYPOTHEKENTILGUNG	262
	** I **	
	IF...THEN	28
	INDIZIERUNG, DOPPELTE	101
	INFORMATIONEN	8
	INFORMATIONSEINGABE	26
	INHALTSVERZEICHNIS	35
	INPUT	26, 98
	INT	25, 87
	** J **	
	JAHRESZAHLEN	232
	JAHRESZAHLEN, HISTORISCHE	232

** K **

KASSETTE	13
KGU	50
KOMMANDO	12, 20
KORREKTUR	24
KREIS	307

** L **

LADEN	34
LAENDER	238, 282
LAUFVARIABLE	32
LEERZEILE	18
LET	23, 98
LINSE	163
LIST	20
LOAD	34
LOG	25
LOOP	32

** M **

MATHEMATIK	39, 291
MENUE	174
MENUETECHNIK	174
MITTEL, ARITHMETISCHES	269
MITTELWERT	27, 269
MULTIPLIKATION	23

** N **

NAME	10
NATION	245
NETTOBETRAG	80
NEW	21
NEXT	32
NORMALGRAPHIK	132

Voß C64/ Schule	Stichwortverzeichnis	Seite 329
** O **		
	OEKOLOGIE	211
	OPEN	35
** P **		
	PENDEL	157
	PENDELBEWEGUNG	157
	PERIODENSYSTEM	122
	PHYSIK	131
	POKE	138
	POTENZIERUNG	23
	PRIMZAHL	59
	PRIMZAHLENPRUEFUNG	59
	PRINT	18
	PRINT#N	35
	PROBLEMANALYSE	14, 41
	PROBLEMVORSTELLUNG	41
	PROGRAMM	41
	PROGRAMMANWEISUNG	11, 12
	PROGRAMMBESCHREIBUNG	41
	PROGRAMMIEREN	16
	PROGRAMMSCHLEIFE	32
	PROGRAMMVERZWEIGUNG	28
	PROZENTRECHNUNG	88
	PUNKT	297, 298
	PYTHAGORAS	41
** R **		
	READ	99
	REAKTION, CHEMISCHE	109
	REAKTIONSGLEICHUNG	109
	RECHENGENAUIGKEIT	92
	RECHENOPERATION	23
	RECHNEN, STOECHIOMETRISCHES	116
	RECHNUNG	18
	REM	37
	RESTORE	102
	RESTORE-TASTE	98, 298
	RETURN	181
	RETURN-TASTE	24
	RND	25
	RUN	19
	RUN/STOP-TASTE	36, 98, 162, 298
	RUNDEN	85

Voß C64/ Schule	Stichwortverzeichnis	Seite 330
	** S **	
	SATZ	17
	SATZNUMMER	28
	SAVE	35
	SCHLEIFE	32
	SCHRITTWEITE	32
	SHIFT	96
	SIN	25
	SINUS	317
	SINUSLINIE	317
	SONDERZEICHEN	9
	SORTIEREN	203
	SORTIERPROGRAMM	207
	SPALTENINDEX	101
	SPEICHER, EXTERNER	13, 34
	SPEICHERN	34
	SPEICHERPLATZ	100
	SPEICHERSTELLE	10
	SPEKTRUM	168
	SPRACHE	179
	SPRUNG	28
	SPRUNG, BEDINGTER	28
	SPRUNG, UNBEDINGTER	28
	SQR	25
	STATEMENT	17
	STEP	33
	STEUERBETRAG	80
	STOECHIOMETRIE	116
	STOP	36
	STRICH	136
	STRING	11, 18
	STRINGVARIABLE	22
	SUBTRAKTION	23
	SYMBOL	9
	** T **	
	TAB	21
	TAN	25
	TASTATUR	13, 96
	** U **	
	UMWELTVERSCHMUTZUNG	223

** U **

VARIABLE	10, 22
VARIABLE, GANZZAHLIGE	22
VARIABLE, INDIZIERTE	100
VARIABLE, REELLE	22
VARIABLENLISTE	41
VARIABLENNAME	18, 22
VC1541	35
VERARBEITUNG	12
VERBEN	182
VERBEN, ENGLISCHE	182
VERBEN, UNREGELMAESSIGE	182
VERZWEIGUNG	28
VOKABELN	189
VOKABELTEST	196

** W **

WACHSTUM	212
WACHSTUM, GEBREMSTES	217
WACHSTUM, UNGEBREMSTES	212
WASSERMOLEKUEL	103
WERT	10
WERTZUWEISUNG	22
WIRTSCHAFT	255
WIRTSCHAFTSKRAFT	282

** Z **

ZAHL	18
ZAHL, EULERSCHE	74
ZEILENINDEX	101
ZENTRALEINHEIT	13
ZIFFER	9
ZINS	256
ZINSRECHNUNG	256

ACHTUNG AUTOREN

SIE haben ein gutes Programm oder ein Manuskript zu einem interessanten Buch geschrieben oder würden dies gerne tun

SIE würden das Ergebnis Ihrer Arbeit gerne in größeren Stückzahlen vermarktet sehen

SIE suchen dafür den leistungsfähigen Verlag und Vertriebspartner Ihres Vertrauens

WIR besitzen große Erfahrung in der professionellen Vermarktung von Software und Literatur, nicht nur in Deutschland, sondern auch weltweit

WIR haben in den letzten 12 Monaten mit dem Verkauf von 200.000 Büchern und 50.000 Programmen unsere Leistungsfähigkeit unter Beweis gestellt

WIR suchen weitere Autoren, mit denen wir gemeinsame Erfolge erringen können

SIE und **WIR** sollten zusammenarbeiten.

Bitte, senden Sie eine Beschreibung Ihres Programms oder Ihres Buchprojekts an Dr. Achim Becker c/o DATA BECKER, oder fordern Sie einfach unsere unverbindlichen „Informationen für Autoren“ an.

IHR GROSSER PARTNER FÜR KLEINE COMPUTER
DATA BECKER

Merowingerstraße 30 · 4000 Düsseldorf · Telefon (02 11) 31 00 10 · im Hause AUTO BECKER

Alles, was Sie schon immer über Ihren **COMMODORE** wissen wollten!



DAS
MASCHINENSPRACHE-
BUCH ZUM
COMMODORE 64
1984, ca. 200 Seiten,
DM 39,-

Eine leicht verständliche Einführung in die Programmierung in Maschinensprache für alle, denen die Programmierung des Commodore 64 in BASIC nicht mehr ausreicht. Sie lernen Aufbau, Arbeitsweise und Register des 6510-Mikroprozessors kennen und werden über Bit, Bytes, Daten und Adressen aufgeklärt. Anhand von Beispielen lernen Sie alle Befehle des 6510 kennen und anwenden. Dabei werden die Analogien zu BASIC Ihnen beim Verständnis helfen. Ein weiteres Kapitel beschäftigt sich mit der Eingabe von Maschinenprogrammen. Dort erfahren Sie auch alles über Monitor-Programme sowie über Assembler. Zum einfachen und komfortablen Erstellen Ihrer eigenen Maschinensprache enthält das Buch einen kompletten Assembler, damit Sie gleich von Anfang an komfortabel und effektiv programmieren können. Weiterhin finden Sie dort einen Disassembler, mit dem Sie sich Ihre Maschinenprogramme oder die Routinen des BASIC-Interpreters und des Betriebssystems ansehen können. Ein besonderer Clou ist ein in BASIC geschriebener Einzelschrittssimulator, mit dem Sie Ihre Programme schrittweise ausführen können. Dabei werden Sie nach jedem Schritt über Registerinhalte und Flags informiert und können den logischen Ablauf Ihres Programms verfolgen. Eine unschätzbare Hilfe besonders für den Anfänger. Als Beispielprogramm finden Sie ausführlich beschriebene Routinen zur Grafikprogrammierung und für BASIC-Erweiterungen. Natürlich sind alle Beispiele und Programme auf Ihren Commodore 64 zugeschnitten.



DAS TRAININGSBUCH
ZUM SIMON's BASIC,
1984, ca. 300 Seiten,
DM 49,-

SIMON's BASIC ist ein Hit — wenn man es richtig nutzen kann. Deshalb gibt es jetzt zu dieser vielseitigen Befehlsenerweiterung unser umfangreiches Trainingsbuch. Auf über 300 Seiten erklärt es Ihnen detailliert den Umgang mit den über 100 Befehlen des SIMON's BASIC. Alle Befehle werden ausführlich dargestellt, auch die, die nicht im Handbuch stehen! Natürlich zeigen wir auch die Masken des SIMON's BASIC und geben wichtige Hinweise, wie man diese umgeht. Zum Buch gehören zahlreiche Beispielprogramme und viele interessante Programmiertricks. Alle Befehle sind entsprechend ihren Anwendungen in entsprechende Kapitel geordnet: Einführung in das CBM BASIC 2.0 — Programmierhilfen — Fehlerbehandlung — Programmschutz — Programmstruktur — Variablen — Zahlenbehandlung — Eingabekontrolle — Ein/Ausgabe Peripheriebefehle — Graphik — Zeichensatzerstellung — Sprites — Musik — Steuernde Peripherie — SIMON's BASIC und die Verträglichkeit mit anderen Erweiterungen und Programmen. Dazu ein umfangreicher Anhang. Nach jedem Kapitel finden Sie Testaufgaben zum optimalen Selbststudium und zur Lernerfolgskontrolle. DAS TRAININGSBUCH ZUM SIMON's BASIC sollte jeder Anwender dieser universellen Befehlsenerweiterung unbedingt haben.

DATA BECKER BÜCHER



DAS GROSSE
FLOPPY-BUCH,
1983, ca. 320 Seiten,
DM 49,-

Darauf haben Sie gewartet: Endlich ein Buch, das Ihnen ausführlich und verständlich die Arbeit mit der Floppy VC-1541 erklärt. DAS GROSSE FLOPPY BUCH ist für Anfänger, Fortgeschrittene und Profis gleichermaßen interessant. Sein Inhalt reicht von der Programmspeicherung bis zum DOS-Zugriff, von der sequentiellen Datenspeicherung bis zum Direktzugriff, von der technischen Beschreibung bis zum ausführlich dokumentierten DOS-Listing, von den Systembefehlen bis zur detaillierten Beschreibung der Programme der Test/Demodiskette. Exakt beschriebene Beispiel- und Hilfsprogramme ergänzen dieses neue Superbuch. Aus dem Inhalt: Der erste Kontakt — Das Speichern von Programmen — Die Floppy-Systembefehle — Sequentielle Datenspeicherung — Relative Datenspeicherung — Die Fehlermeldungen und Ihre Ursachen — Der Direktzugriff — Der Zugriff auf das DOS — Technik der Floppy und der Diskette — DOS-Listing der VC-1541 — Dienstprogramme — BASIC-Erweiterungen und Programme zur komfortablen Benutzung der VC-1541 - Overlaytechnik — Diskmonitor — IEC-Bus und serieller Bus — Gemeinsamkeiten mit den großen CBM-Floppies und Unterschiede gegenüber der VC-1541. Mit dem GROSSEN FLOPPY-BUCH meistern Sie auch Ihre Floppy.



64 FÜR PROFIS
1983, ca. 280 Seiten,
DM 49,-

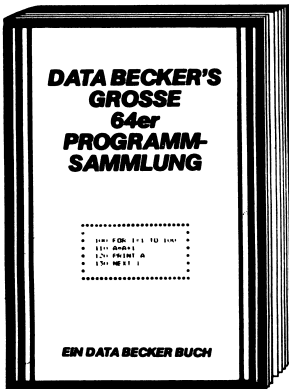
Wer besser und leichter in BASIC programmieren möchte, der braucht dieses Buch. 64 FÜR PROFIS zeigt, wie man erfolgreich Anwendungsprobleme in BASIC löst und verrät die Erfolgsgeheimnisse der Programmierprofis. Vom Programm-entwurf über Menüsteuerung, Maskenaufbau, Parameterisierung, Datenzugriff und Druckausgabe bis hin zur guten Dokumentation wird anschaulich mit Beispielen dargestellt, wie Profiprogrammierung vor sich geht. Besonders stolz sind wir auf die völlig neuartige Datenzugriffsmethode QUISAM, die in diesem Buch zum ersten Mal vorgestellt wird. QUISAM erlaubt eine beliebige Datenzusatzlänge, die dynamisch mit der Eingabe der Daten wächst. Eine lauffertige Literaturstellenverwaltung veranschaulicht die Arbeitsweise von QUISAM. Neben diesem Programm finden Sie auch noch weitere Programme zur Lager- und Adressenverwaltung, Textverarbeitung und einen Reportgenerator. Alle diese Programme sind mit Variablenliste versehen und ausführlich beschrieben. Damit sind diese für Ihre Erweiterungen offen und können von Ihnen an Ihre speziellen Bedürfnisse angepaßt werden. Mit 64 FÜR PROFIS steigen Sie in die Welt der Programmierprofis ein.

MIT DATA BECKER BÜCHERN MACHEN

SIE MEHR AUS IHREM COMMODORE



Das sollte Ihr erstes Buch zum COMMODORE 64 sein: 64 FÜR EINSTEIGER ist eine sehr leicht verständliche Einführung in Handhabung, Einsatz, Ausbaumöglichkeiten und Programmierung des COMMODORE 64, die keinerlei Vorkenntnisse voraussetzt. Sie reicht vom Anschluß des Geräts über die Erklärung der einzelnen Tasten und Funktionen sowie die Peripheriegeräte und ihre Bedienung bis zum ersten Befehl. Schritt für Schritt führt das Buch Sie in die Programmiersprache BASIC ein, wobei Sie nach und nach eine komplette Adressverwaltung erstellen, die Sie anschließend nutzen können. Zahlreiche Abbildungen und Bildschirmfotos ergänzen den Text. Viele Anwendungsbeispiele geben nützliche Anregungen zum sinnvollen Einsatz des COMMODORE 64. Das Buch ist sowohl als Einführung als auch als Orientierung vor dem 64er Kauf gut geeignet. 64 FÜR EINSTEIGER 1984, ca. 200 Seiten, DM 29,—



Diese neue, umfangreiche Programmsammlung hat es in sich. Über 50 Spitzenprogramme für den COMMODORE 64 aus den unterschiedlichsten Bereichen, von attraktiven Superspielen („Senso“, „Pengo“, „Seeschlacht“, „Poison Square“, „Memory“) über Graphik- und Soundprogramme („Fourier 64“, „Akustograph“, „Funktionsplotter“) sowie Utilities („SORT“, „Renumber“, „Disk Init“, „Menue“) bis hin zu kompletten Anwendungsprogrammen wie „Videothek“, „File Manager“ und einer komfortablen Haushaltsbuchführung, in der fast professionell gebucht wird. Der Hit sind zu jedem Programm aktuelle Programmertips und Tricks der einzelnen Autoren zum Selbermachen. Also — nicht nur abtippen, sondern auch dabei lernen und wichtige Anregungen für die eigene Programmierung sammeln.

DATA BECKER'S GROSSE 64er PROGRAMMSAMMLUNG 1984, ca. 250 Seiten, DM 49,—



Achtung Hobbyelektroniker: Dieses neue Buch enthält nicht nur alles über die Ausbaumöglichkeiten des COMMODORE 64 und wie man ihn über seine Schnittstellen mit dem „Rest der Welt“ verbindet, sondern auch umfassende Informationen über die vielfältigen Einsatzmöglichkeiten des COMMODORE 64 von der Lichtorgel über Motorsteuerung, Spannungs- und Temperaturmessung bis zur programmierbaren Stromversorgung, und wie man diese verwirklicht. Umfassende Darstellung der Grundlagen. Als Clou enthält das Buch zehn komplette Schaltungen zum Selberbauen, vom Eprommer über Eprom-Karte, Logic Analyzer, Frequenzzähler, Hardware-Tracer, Pulsmeßgerät, Klatschschalter und Digital-Voltmeter bis zur preiswerten Spracheingabe-Sprachausgabe. Jeweils komplett mit Schaltplan, Layout und Softwarelisting. Auch für Profis interessant.

DER COMMODORE 64 UND DER REST DER WELT 1984, ca. 220 Seiten, DM 49,—



Der COMMODORE 64 ist ein Musikgenie. Mit diesem Buch lernen Sie alles über seine musikalischen Fähigkeiten und wie Sie diese programmtechnisch nutzen. Der Inhalt des Musikbuches reicht von einer Einführung in die Computermusik über die Erklärung der Hardware-Grundlagen und die Programmierung in BASIC bis hin zur fortgeschrittenen Musikprogrammierung in Maschinensprache. Zahlreiche Beispielprogramme, komplette Songs und nützliche Routinen ergänzen den leicht verständlichen Text. Geschrieben wurde das Buch von Thomas Dachselt, dem Autor der weltbekannten Musikprogramme SYNTHIMAT und SYNTHESOUND. Erschließen Sie sich die Welt des Sounds und der Computermusik mit dem Musikbuch zum COMMODORE 64.

DAS MUSIKBUCH ZUM
COMMODORE 64
1984, ca. 200 Seiten, DM 39,—



So etwas haben Sie gesucht: Umfassendes Nachschlagewerk zum COMMODORE 64 und seiner Programmierung. Allgemeines Computerlexikon mit Fachwissen von A-Z und Fachwörterbuch mit Übersetzungen wichtiger englischer Fachbegriffe - das DATA BECKER LEXIKON ZUM COMMODORE 64 stellt praktisch drei Bücher in einem dar. Es enthält eine unglaubliche Vielfalt an Informationen und dient so zugleich als kompetentes Nachschlagewerk und als unentbehrliches Arbeitsmittel. Viele Abbildungen und Beispiele ergänzen den Text. Ein Muß für jeden COMMODORE 64 Anwender.

DAS DATA BECKER
LEXIKON ZUM
COMMODORE 64
1984, ca. 350 Seiten, DM 49,—



Graphik ist eine der Hauptstärken des COMMODORE 64. Mit diesem leicht verständlich geschriebenen, aber sehr umfassenden neuen Buch lernen Sie, wie Sie die graphischen Fähigkeiten für Ihre eigenen Programme optimal nutzen. Der Inhalt reicht von den Grundlagen der Graphikprogrammierung über das Erzeugen einfacher Figuren, die Arbeit mit Sprites, Zeichensatzprogrammierung, Hardcopy und IRQ-Handhabung bis hin zur Funktionendarstellung, Laufschrift, Statistik, 3-D, CAD, den Geheimnissen der Actionspiele und Lightpenanwendungen. Zahlreiche Beispielprogramme, nützliche Routinen und komplette Anwendungen ergänzen diese Buch, das die faszinierende Welt der Computergraphik jedermann zugänglich macht. Geschrieben wurde das Graphikbuch von dem bekannten SUPERGRAPHIK-Autor Axel Plenge.

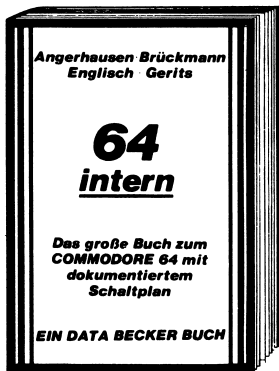
DAS GRAFIKBUCH
ZUM COMMODORE 64 1984, ca. 250 Seiten, DM 39,—

DATA BECKER BÜCHER



64 TIPS & TRICKS,
2. Auflage,
1983, ca. 290 Seiten,
DM 49,-

64 Tips & Tricks ist eine hochinteressante Sammlung von Anregungen zur fortgeschrittenen Programmierung des COMMODORE 64, Poke's und anderen nützlichen Routinen, interessanten Programmen sowie aktuellen Programmertips und -tricks. Aus dem Inhalt: 3D-Graphik in BASIC — Farbige Balkengraphik — Definition eines eigenen Zeichensatzes — Die Tastaturbelegung und ihre Änderung — Dateneingabe mit Komfort — Simulation der Maus mit einem Joystick — BASIC für Fortgeschrittene — Ihr COMMODORE 64 spricht Deutsch — CP/M auf dem COMMODORE 64 — Druckeranschluß über den USER-Port — Datenübertragung von und zu anderen Rechnern — Der Expansionsport — Synthesizer in Stereo — Retten einer nicht ordnungsgemäß geschlossenen Datei — Erzeugen einer BASIC-Zeile in BASIC — Der Kassettenspeicher als Datenspeicher — Sortieren von Stringfeldern — Multitasking auf dem COMMODORE 64 — POKE's und die Zeropage — GOTO, GOSUB und RESTORE mit berechneten Zeilennummern INSTR und STRING-Funktion — Repeatfunktion für alle Tasten — und vieles andere mehr. Alle Maschinenprogramme mit BASIC-Ladeprogrammen. 64 TIPS & TRICKS ist eine echte Fundgrube für jeden COMMODORE 64 Anwender.



64 INTERN,
3. Auflage,
1983, ca. 320 Seiten,
DM 69,-

64 INTERN ist unser großes Buch zum COMMODORE 64. Ideal für alle, die sich näher mit Programmierung, Technik und Betriebssystem von COMMODORE's Supermaschine auseinandersetzen wollen. Detailliert werden Architektur und technische Möglichkeiten des C-64 beschrieben, vom Prozessor 6510 über Video-Controller 6569, Sound Controller 6581, Ein-/Ausgabesteuerung mit CIA 6526 bis hin zum seriellen IEC-Bus, USER-Port und EXPANSION-PORT. Parallel dazu wird die Programmierung der hochauflösenden Graphik und der Sprites erklärt, die Sonderprogrammierung und die Programmierung des C-64 in Maschinensprache. Mit einem ausführlich dokumentierten ROM-Listing wird das Betriebssystem zerlegt. Wertvolle Hinweise zum Umsetzen von Programmen bringt ein Vergleich zwischen VC-20, C-64 und CBM 8000. Zahlreiche lauffertige Beispielprogramme und Routinen, z. B. Graphik-Aid, HARDCOPY, RENEN und PRINT USING runden das Buch ab. Der Clou von 64 INTERN sind neben den zahlreichen Schaltbildern und Blockdiagrammen 2 Original COMMODORE-Schaltpläne zum Ausklappen, die zusätzlich sehr ausführlich beschrieben und dokumentiert sind. Dieses Buch sollte jeder COMMODORE 64- Anwender und Interessent haben.

MIT DATA BECKER BÜCHERN MACHEN

SIE MEHR AUS IHREM COMMODORE

DATA BECKER BÜCHER



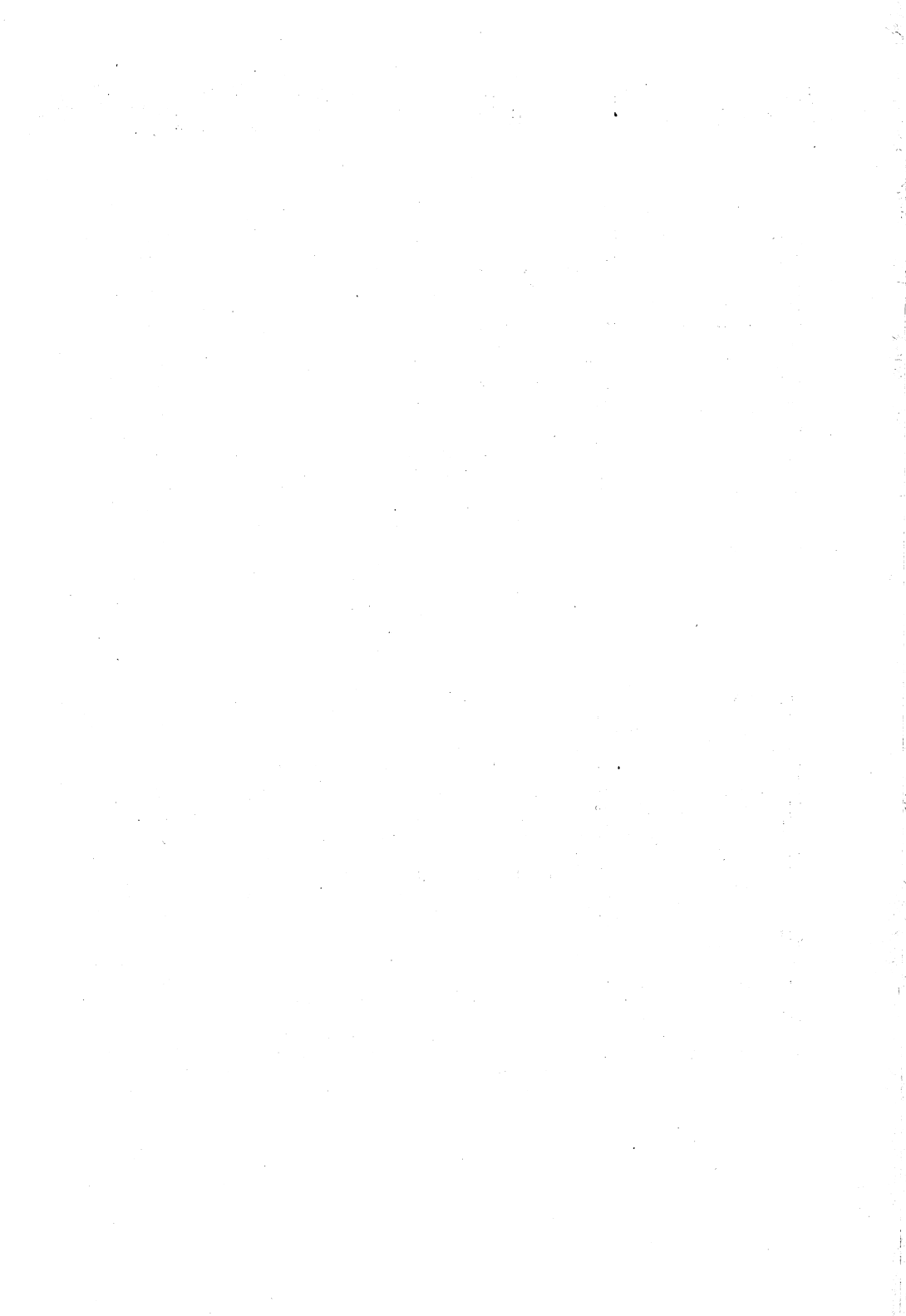
VC-20 INTERN,
2. Auflage,
1983, ca. 230 Seiten,
DM 49,-

Die bereits 3. Auflage von VC-20 INTERN ist wieder erheblich erweitert worden. Das Buch beschäftigt sich ausführlich mit der Technik und dem Betriebssystem des VC-20. Dazu gehört natürlich zuerst einmal ein komplettes und ausführlich dokumentiertes ROM-Listing. Dazu gehört auch die Belegung der ZEROPAGE, dem wichtigsten Speicherbereich für den 6502-Prozessor, eine übersichtliche Auflistung der Adressen aller Betriebssystemroutinen, ihrer Bedeutung und ihrer Übergabeparameter. Dies ermöglicht dem Programmierer endlich, den VC-20 von Maschinensprache aus sinnvoll einzusetzen. Denn warum Routinen, die bereits vorhanden sind, noch einmal schreiben? Für Einsteiger befindet sich eine Einführung in die Programmierung in Maschinensprache am Anfang des Buches. Diese behandelt die Themen Maschinensprache-monitor, Disassembler und Assembler. Außerdem wird die Verbindung von BASIC- und Maschinenspracheprogrammen besprochen. Doch nicht nur die Software, auch die Hardware wird ausführlich beschrieben. Detailliert werden alle wichtigen IC's im VC-20 in Ihrer Arbeitsweise beschrieben. Neben einem übersichtlichen Blockschaltbild enthält VC-20 INTERN noch als besonderen Clou drei Original COMMODORE Schaltpläne zum Ausklappen. Dieses Buch braucht jeder, der sich näher mit Technik und Maschinenspracheprogrammierung des VC-20 auseinandersetzen möchte.



VC-20 TIPS & TRICKS,
2. Auflage,
1983, ca. 230 Seiten,
DM 49,-

Die überarbeitete und erheblich erweiterte 2. Auflage von VC-20 TIPS & TRICKS enthält endlich eine genaue Erklärung des Speicheraufbaus des VC-20 in allen möglichen Erweiterungsstufen. Vergessen Sie alles, was Ihnen über Grafik und Sound auf dem VC-20 bisher geliefert wurde. In diesem Buch finden Sie von den Grundlagen angefangen über die programmierbaren Zeichen, über eine Grafikerweiterung in Maschinensprache, über luxuriöse Programme wie z. B. Grafikeditor, Funktionenplotter oder 3D-Grafiken bis hin zur Supererweiterung zum VC-20 alles Wissenswerte und zum Arbeiten Notwendige für eine gute Grafik. Ebenso ausführlich wird die Tonerzeugung behandelt. Neben der Erläuterung der Grundlagen und einem Soundeditor sind natürlich auch fertige Programme, z. B. der VC-20 als Synthesizer oder Schlagzeug abgedruckt. Auf über 100 Seiten finden Sie viele nützliche Tricks, POKE-Befehle, BASIC-Erweiterungen fix und fertig zum Eintippen. Dazu gehören z. B. Unnew, automatische Zeilennummerierung, die Programmierung des User-Port, ein Diskmenue, Hardcopy und so weiter. Bringen Sie Ihrem VC-20 völlig neue Befehle bei! Abgerundet wird das Buch durch Anwenderprogramme zum Eintippen, wie Dateiverwaltung, Textverarbeitung und Spiele. VC-20 TIPS & TRICKS braucht jeder VC-20 Besitzer, der mehr über seinen Computer wissen möchte.



DAS STEHT DRIN:

Besonders für Schüler der Mittel- und Oberstufe geschrieben, enthält das Buch viele interessante Problemlösungs- und Lernprogramme, die besonders ausführlich und leicht verständlich beschrieben sind. Sie ermöglichen ein intensives Lernen mit viel Spaß, unter anderem mit folgenden Themen:

- Satz des Pythagoras
- geometrische Reihen
- Mechanische Hebel
- Exponentielles Wachstum
- unregelmäßige Verben
- quadratische Gleichungen
- Pendelbewegungen
- Molekülbildung
- Vokabeln lernen
- Zinseszinsrechnung

Ein kurzer Überblick über die Grundlagen der EDV, eine knappe Wiederholung der wichtigsten BASIC-Elemente und eine Einführung in die Grundzüge der Problemanalyse vervollständigen das Ganze.

UND GESCHRIEBEN HAT DIESES BUCH:

Werner Voß ist Professor für Statistik an der Universität Bochum. Zahlreiche Veröffentlichungen im Bereich Statistik und Datenverarbeitung.