VOSS

DAS SCHULBUCH ZUM CPC

EIN DATA BECKER BUCH

VOSS

DAS SCHULBUCH ZUM CPC

EIN DATA BECKER BUCH

ISBN 3 - 89011 - 040 - 1

Copyright © 1985 DATA BECKER GmbH
Merowingerstraße 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 Patentlage mitgeteilt. Sie sind ausschließlich für Amateur- und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden.

Alle Schaltungen, technischen Angaben und Programme in diesem Buch wurden von dem 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 ist der Autor jederzeit dankbar.

Vorwort

Über den Einsatz des Computers in der Schule wird sehr viel geredet, aber sehr wenig getan. DATA BECKER hat sich daher entschlossen, selbst eine Buchreihe zum Thema herauszugeben. Ein sehr komfortables BASIC und hervorragende technische Leistungen prädestinieren den CPC 464 für eine Aufnahme in diese Buchreihe.

Werner Voß, Professor für Statistik an der Universität Bochum, der nebenberuflich im Bereich der Datenverarbeitung und Programmierung tätig ist, 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 eingeführt.

Es gibt kaum ein Unterrichtsfach, in dem sich der CPC 464 nicht gewinnbringend einsetzen läßt. Viel Spaß dabei!

Dr. Achim Becker

GLIEDERUNG
==========

		Seite
ø.	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
1.1ø	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	Würfel	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-Programme	136
4.4	Die Federwaage	144
4.5	Der Satz des Archimedes	151
4.6	Pendelbewegung	157
4.7	Optische Abbildung	163
4.8	Kräfteparallelogramm	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 Bundesländer 282

9. Mathematik II 291

9.1 Vorbemerkung 291
9.2 Gitter 293
9.3 Die Gerade 296
9.4 Der Kreis 299
9.5 Die Sinuslinie 3o3

Seite

9.6 Mathem. Tafel 1 ..... 3o6
9.7 Mathem. Tafel 2 ..... 3o9
9.8 Mathem. Tafel 3 313
9.9 Ellipse 316
9.1o Gauß'sche Normalverteilung 32o
9.11 Regression 324
9.12 Permutationen 332
9.13 Kombinationen 336
9.14 Lotto 342
9.15 Binomialverteilung 347
9.16 Zentrales Grenzwerttheorem 358
9.17 Wahlen 376

1o. Ausblick 381

Stichwortverzeichnis 383

Vorwort

Computer - wie der SCHNEIDER CPC 464 - 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 SCHNEIDER CPC 464, dazu einsetzen kann, wesentliche Probleme aus dem

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 erprobt.

Im Bereich des Programmierens und des Rechnereinsatzes 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. Aufgaben aus der Mathematik
2. Probleme aus der Chemie
3. Physikalische Aufgabenstellungen
4. Aufgaben aus dem Sprachunterricht
5. Aufgaben aus dem Bereich Biologie/Ökologie
6. Probleme aus Erdkunde und Geschichte
7. Wirtschaftsprobleme
8. Sonstige Aufgabenstellungen

Diesen inhaltlichen Kapiteln werden einige Ausführungen zu den Grundlagen und Bedingungen des Rechnereinsatzes beigefügt, 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").

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 programmiertechnische 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.

Bei komplizierteren Problemen in den späteren Kapiteln wird dann allerdings auch der Einsatz weiterer Sprachelemente erforderlich, die dann erst von Fall zu Fall besprochen werden.

Weiterhin möge der Leser insbesondere beachten, daß manche der vorgestellten Programme nur exemplarischen Charakter haben:

Wenn z.B. in einem Vokabel-Abfrageprogramm nur 1Ø 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.

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.

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:

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 Ø 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.

Feld: Eine Folge von Symbolen (z.B. Worte oder Zahlen) bilden ein Feld.

Beispielsweise besteht das Wort PRINT aus 5 Symbolen, belegt mithin 5 byte im Rechner, und da diese 5 Symbole zusammengehören, bilden sie ein Feld.

Entsprechendes gilt für die Zahl 178. Sie wird in einem dreistelligen Feld gespeichert (man sieht, daß es auch einstellige Felder geben kann).

Variable: Eine Variable ist eine veränderliche Größe oder - präziser gesprochen - ein Name, unter dem verschiedene Werte (s.u.) angesprochen werden können.
Beispielsweise stehen hinter dem Variablennamen "Körpergröße" die Werte 178, 185, 167 usw., hinter dem Namen "Kinderzahl" z.B. die Werte 1, 2, Ø, 4 usw., hinter dem Namen "Ort" z.B. die Werte "Dortmund", "Bochum", "Kassel" und dergl.

Wert: Wie aus der vorangegangenen Begriffsklärung schon zu entnehmen ist, verstehen wir unter dem Begriff "Wert" eine bestimmte Ausprägung 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.

Die Werte belegen jeweils ein Feld im Rechner (s.o.). Man spricht auch von einer Speicherstelle.

String: Einen Text, der z.B. als Variablenausprägung auftreten kann (s.o.), nennt man "string"; allgemein gesprochen handelt es sich um eine Zeichenkette.

Array: Die Gesamtheit aller Ausprägungen einer Variablen (Zahlen oder Worte) nennt man "array".

Datensatz: 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 Programmanweisung:

```
PRINT  "OTTO", "SUSI", "OLAF"
```

Der Datensatz wird auch als "record" bezeichnet.

Datei: 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 Programmanweisungen eines Programms insgesamt betrachten.

Die Datei wird auch als "file" bezeichnet.

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)
2. Informationsspeicherung
3. Verarbeiten von Daten im engeren Wortsinn (z.B. Rechnen oder Umstellen von Texten und dergl.)
4. Ausgabe von Ergebnissen

E	Eingabe
V	Verarbeitung
A	Ausgabe

Informationseingabe, -verarbeitung und -ausgabe machen 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.)

Entsprechend den unterschiedlichen Teilaufgaben, die der Rechner erledigen muß, unterteilen wir:

1. Eingabegeräte (z.B. Tastatur)
2. Verarbeitungsbereich (die sog. Zentraleinheit)
3. Ausgabegeräte (Bildschirm oder z.B. Drucker)

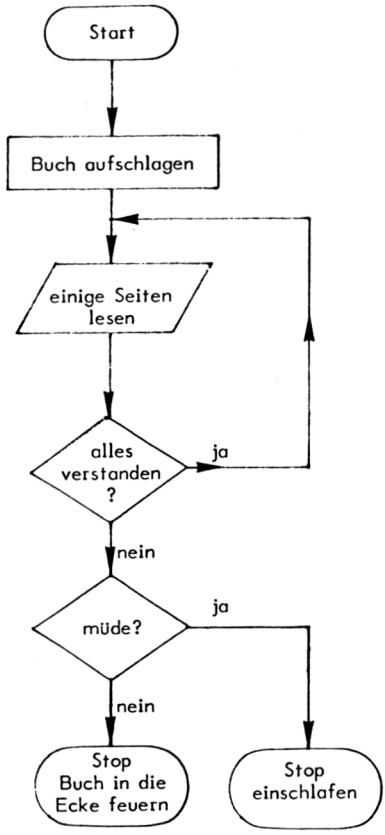
Darüber hinaus gibt es sog. externe (oder periphere) Speicher. Sie dienen dazu, Programme oder Datenbestände auf Dauer aufzubewahren. In der Regel handelt es sich dabei um Tonband-Kassetten oder um sog. Disketten, die in entsprechenden Laufwerken bespielt bzw. gelesen werden können.

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.

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.

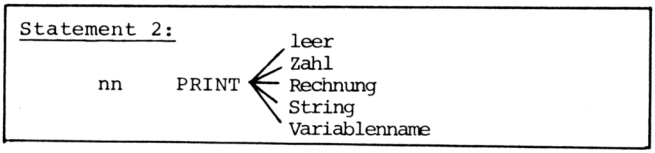
Regel 1:	Jeder Satz in BASIC benötigt eine Satznummer
Regel 2:	Jeder Satz muß mit der ENTER-Taste abgeschlossen werden.
Regel 3:	In einem Satz können mehrere Statements stehen, die dann durch Doppelpunkte voneinander zu trennen sind.

Statement 1:

nn END

Das END-Statement beschließt das BASIC-Programm.

Statement 2:



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.

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
```

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 Ergebnisausdrucke, die am linken Bildschirmrand beginnen. Will man dies ändern, kann man sich der TAB-Funktion (TAB = Tabulator) bedienen.

```
PRINT TAB(1ø) 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 Dezimalkommas 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 = Zahl / Rechnung / String / 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 ENTER-Taste zu schließen und ihn darunter einfach neu zu schreiben. Soll ein Satz ganz gelöscht werden, so gibt man einfach seine Satznummer ein plus ENTER-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)
```

Wichtige Funktionen

Name	Aufgabe
SIN COS TAN	berechnet die Winkelfunktionswerte, wobei als Argument Einheiten des Kreisparameters π zu verwenden ist
LOG EXP	natürlicher Logarithmus Umkehrung dazu
SQR	Quadratwurzel
ABS	Absolutwert
INT	größter, in einem reellen Wert vorhandener, 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.

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 Variablenliste 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 1o:	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.

```
10 CLS
20 INPUT"BITTE WERT EINGEBEN : ";X
30 N=N+1
40 S=S+X
50 INPUT"NOCH EIN WERT (J/N) ";A$
60 IF A$="J" THEN 20
70 AM=S/N
80 PRINT"MITTELWERT = ";AM
90 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.

"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 1Ø mal "GUTEN TAG" untereinander:

```
1Ø FOR I = 1 TO 1Ø
2Ø ?"GUTEN TAG"
3Ø NEXT I
4Ø ?
5Ø ?"ENDE"
6Ø 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"
```

Nach der Aufforderung des Rechners ist dann das Kassettengerät zu bedienen, wobei darauf zu achten ist, daß die eingelegte Kassette nicht so weit vorgespult ist, daß der Anfang des gewünschten Programms nicht mehr gefunden werden kann.

Zum Speichern eines Programms auf Kassette benötigen wir:

Kommando 5:

```
SAVE  "Name des Programms"
```

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

Kommando 6:

```
CAT
```

Auf dem Bildschirm wird dann, wenn der Benutzer den Anweisungen des Rechners gefolgt ist, ein Kassetten-Inhaltsverzeichnis ausgegeben.

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 Anweisung:

Statement 9:	
nn	CLS

Häufig ist es sinnvoll, einen Programmlauf zu stoppen, beispielsweise dann, wenn man eine sog. Endlosschleife programmiert hat. Betätigt man die Taste ESC , so hält die Programmabarbeitung an und wird durch Drücken irgendeiner anderen Taste wieder aufgenommen.
Drückt man die ESC -Taste zweimal, bleibt das Programm endgültig stehen.

Schließlich ist anzumerken, daß mit dem Statement

Statement1o:	nn	STOP

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

Kommando 7:	CONT

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:

Statement 11:

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.

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 denjenigen Aufgabenstellungen orientieren, wie sie zum Beispiel in der gymnasialen Mittelstufe, also in den

Jahrgangsstufen 8 bis 1o 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 Programme in diesem und in den folgenden Kapiteln gilt die folgende wichtige Anmerkung :

> Bei den Programmen wird darauf verzichtet, Eingeabefehler 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, den 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 gegliedert werden.

Wir werden jeweils die folgenden Arbeitsschritte vollziehen :

Arbeitsschritte :

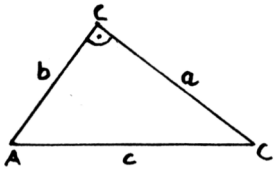
1. SCHRITT : Vorstellung des Problems
2. SCHRITT : Problemanalyse
3. SCHRITT : Flußdiagramm
4. SCHRITT : Programm
5. SCHRITT : Variablenliste
6. SCHRITT : Programmbeschreibung
7. SCHRITT : Ergebnisse

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 (Hypothenuse) 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 bei allen folgenden Problemen auch, daran erinnern, daß ganz allgemein ein Datenverarbeitungsprozeß nach dem schon vorgestellten

E V A - Prinzip

abläuft.

E	=	Eingabe
V	=	Verarbeitung
A	=	Ausgabe

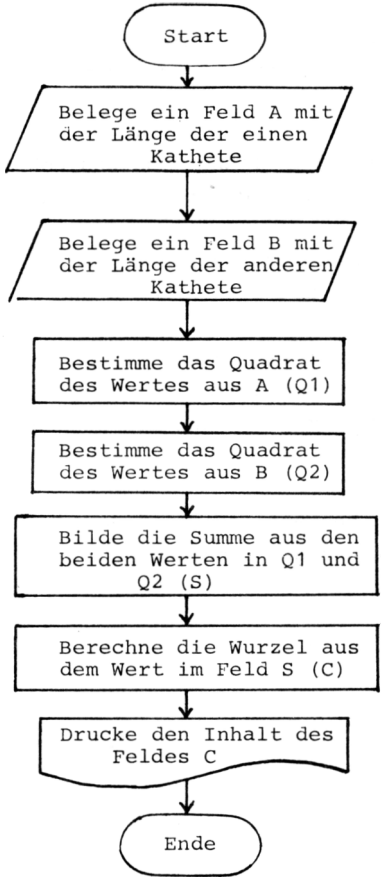
Wir müssen bei der Problemanalyse also zunächst festlegen, welche Informationen der Rechner von uns bekommen muß, damit das gestellte Problem gelöst werden kann.

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

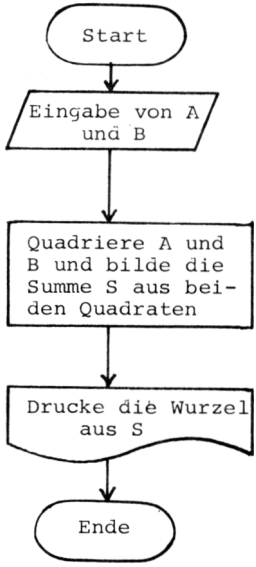
Im Verarbeitungsschritt sind diese beiden Werte zu quadrieren; es ist die Summe dieser beiden Quadrate zu bilden und aus dieser Summe muß dann die Quadratwurzel gezogen werden.

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

3. SCHRITT : Flußdiagramm



Kürzere Version des Flußdiagramms :



4. SCHRITT : Programm

```
10 REM M1-PYTHAGORAS
20 CLS
30 PRINT"PROGRAMM ZUR BESTIMMUNG DER HYPOTHENUSE"
40 PRINT"      EINES RECHTWINKLIGEN DREIECKS."
50 PRINT:PRINT:PRINT TAB(10)"PROF.DR.W.VOSS, 1984"
:PRINT
60 INPUT"ERSTE KATHETE    : ";A
70 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

```
1ø bis 5ø  :  wie oben
6ø INPUT "ZWEI KATHETENWERTE : ";A,B
7ø C=SQR(A*A+B*B)
8ø 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

(6.) SCHRITT : Programmbeschreibung

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

Satz 1ø : Kommentar

Satz 2ø : Der Bildschirm wird freigemacht

Satz 3ø-5ø: Titelausdruck auf dem Bildschirm

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

Satz 7ø : Entsprechend zweite Kathete für Feld B

Satz 8ø : Berechnung des ersten Kathetenquadrats

Satz 9ø : Berechnung des zweiten Kathetenquadrats

Satz 1øø : Bestimmung der Summe der beiden Katheten-quadrate

Satz 11ø : Bestimmung der Wurzel aus der Quadratsumme

Satz 12ø : Ausdruck des Ergebnisses auf dem Bildschirm

Satz 13ø : Beendigung des Programms

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 Hypothenuse rechtwinkliger Dreiecke auszurechnen. In diesem Sinn ist es also ganz allgemein gehalten.

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, zumindest wenn sie so einfach sind wie der Lehrsatz des Pythagoras.

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 Z1 = 3ø und Z2 = 4ø, so ist der g.g.T. offenbar 1ø und das k.g.V. hat den Wert 12ø. (Der Leser möge dies durch Kopfrechnen überprüfen).

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 3ø und 4ø, wie sinnvollerweise vorzugehen ist, wenn man den g.g.T. finden will :

1. Schritt : 4ø : 3ø = 1 Rest 1ø

2. Schritt : 3ø : 1ø = 3 Rest ø

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 :

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

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.

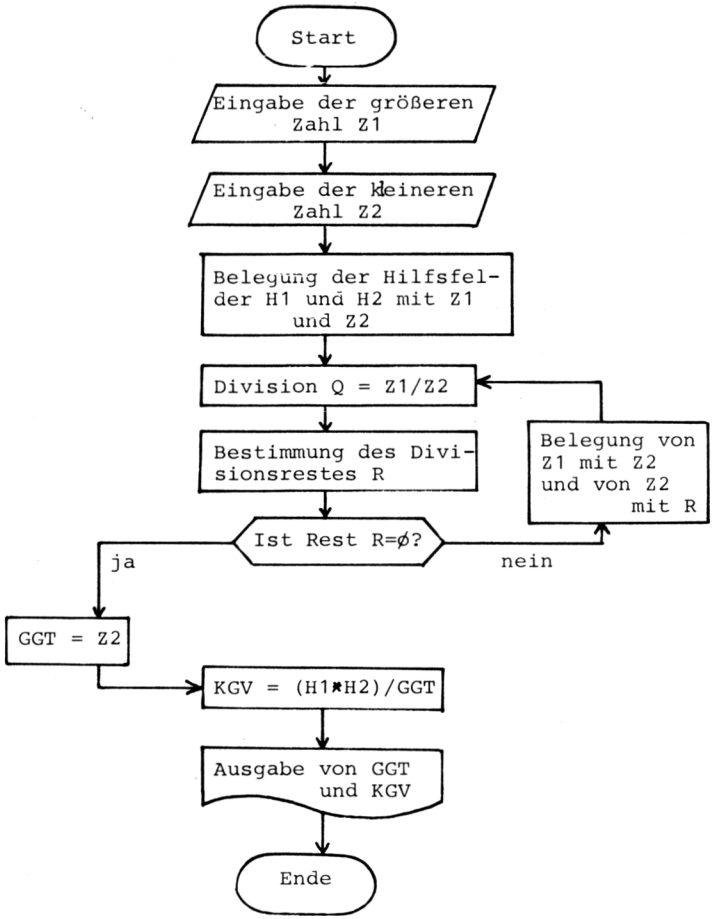
Für die beiden Zahlen Z1 = 78 und Z2 = 42 gilt also :

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

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

der Berechnung des g.g.T. die einzugebenden Werte Z1 und Z2 in zwei Hilfsfeldern H1 und H2 "zwischengespeichert" werden, um für die Berechnung des k.g.V. dann zur Verfügung zu stehen.

5. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM M2-GGT UND KGV
20 CLS
30 PRINT"PROGRAMM ZUR BESTIMMUNG DES G.G.T. UND"
40 PRINT"DES K.G.V. AUS ZWEI EINGEGEBENEN ZAHLEN"
50 PRINT:PRINT:PRINT TAB(10)"PROF.DR.W.VOSS, 1984"
: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	:	Quctient aus beiden Zahlen Z1 und Z2
R	:	Divisionsrest
Z1	:	Größere Zahl
Z2	:	Kleinere Zahl

6. SCHRITT : Programmbeschreibung

Satz 1ø-5ø : Überschrift und Erläuterungen

Satz 6ø : Eingabe der einen Zahl

Satz 7ø : Zwei Leerzeilen

Satz 8ø : Eingabe der zweiten Zahl

Satz 9ø : Belegung der beiden Hilfsfelder H1 und H2 mit den eingegebenen Werten, damit diese später noch zur Verfügung stehen (siehe 14ø)

Satz 1øø : Bestimmung des Quotienten aus beiden eingegebenen Zahlen

Satz 11ø : Bestimmung des Divisionsrestes durch Benutzung der INT-Funktion

Satz 12ø : 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 1øø.

Satz 13ø : Ist hingegen der Rest gleich null, so steht im Feld Z2 der g.g.T., der in das Feld GGT übertragen wird

Satz 14ø : 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 15ø-16ø : Ausdruck der Ergebnisse mit einigen Leerzeilen dazwischen

Satz 17ø : 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-

spielsweise den Wert 78 ein, so fordert das Programm daraufhin die zweite Zahl an :

```
KLEINERE ZAHL      : ?
```

Geben wir zum Beispiel die Zahl 42 ein, so bringt der Rechner die folgenden Ergebnisse auf den Bildschirm :

```
G.G.T. = 6

K.G.V. = 546

ENDE DER BERECHNUNG
```

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 ø

===

25 ist keine Primzahl.

Beispiel : Ausgangszahl X = 7

1. Teiler : 2 7/2 = 3 Rest 1
2. Teiler : 3 7/3 = 2 Rest 1

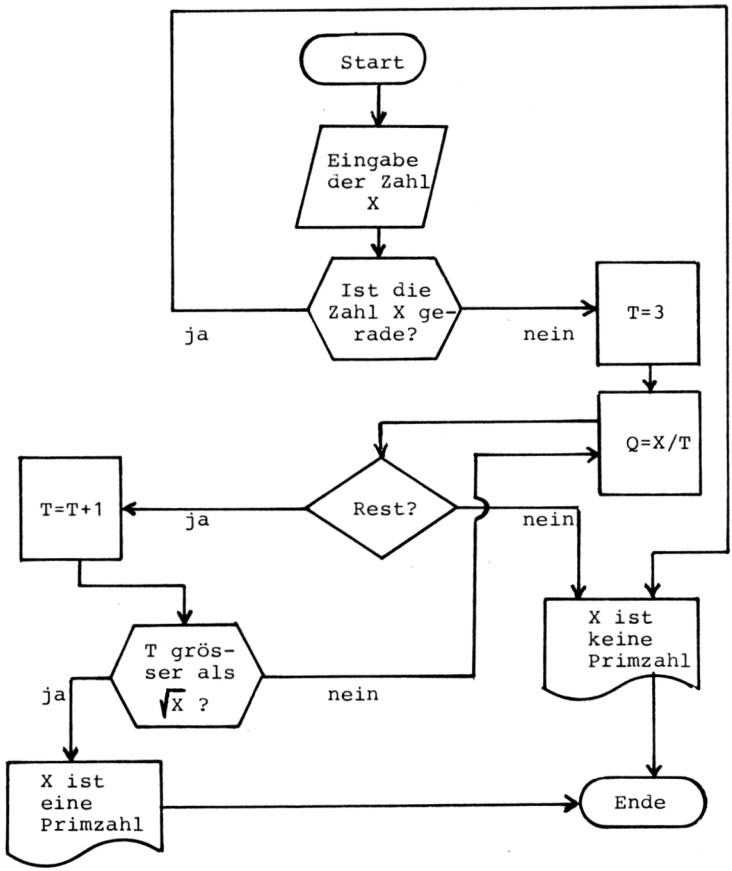
Ein weiterer Teiler braucht nicht mehr probiert zu werden, weil das Divisionsergebnis nicht mehr ganzzahlig werden kann.

Da bei diesem Beispiel kein Rest ø aufgetreten ist, gilt also :

7 ist eine Primzahl.

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.

3. SCHRITT : Flußdiagramm



Dieses Beispiel zeigt übrigens sehr deutlich, daß bei sorgfältiger gedanklicher Durchdringung des Problems das Programm wesentlich vereinfacht und damit auch beschleunigt werden kann :

Man kann schon von vornherein feststellen, daß eine gerade 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.

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.

Der Leser überlege sorgfältig, warum diese Überlegung richtig ist.

(4.) SCHRITT : Programm

```
10 REM M3-PRIMZAHL
20 CLS
30 PRINT"PROGRAMM ZUR PRUEFUNG, OB EINE EINGEGE-"
40 PRINT"BENE ZAHL EINE PRIMZAHL IST ODER NICHT."
50 PRINT:PRINT:PRINT TAB(10)"PROF.DR.W.VOSS, 1984"
: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:PRINT X;" IST EINE PRIMZAHL.":GOTO
 150
140 PRINT:PRINT:PRINT X;" IST KEINE PRIMZAHL, SOND
ERN"
145 PRINT"DURCH ";T;" 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

(6.) SCHRITT : Programmbeschreibung

Satz 1ø-5ø : Überschrift, Räumung des Bildschirms und Titelausdruck auf dem Bildschirm.

Satz 6ø : Anforderung der zu prüfenden Zahl.

Satz 7ø : Belegung des Teilerfeldes T mit 2.

Satz 8ø : Wenn X durch 2 ohne Rest teilbar sein sollte (dann ist X/2 = INT(X/2)), dann weiter bei Satz 14ø; wenn nicht, weiter bei Satz 9ø.

Satz 9ø : Division von X durch T (ergibt Q).

Satz 1øø : Wenn bei der Division kein Rest aufgetreten ist (dann ist Q = INT(Q)), dann weiter bei Satz 14ø, andernfalls weiter bei Satz 11ø.

Satz 11ø : Erhöhung des Teilers T um 1.

Satz 12ø : Solange nach dieser Erhöhung T noch kleiner ist als $\sqrt{X}$, geht es zurück zum Satz 9Ø, andernfalls geht es bei 13ø weiter.

Satz 13ø : Wird der Satz 13ø 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

erfolgt ein Sprung zum Satz 15ø.

Satz 14ø und 145 : Satz 14ø 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 15ø : 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.
```

Geben wir hingegen die Zahl 27 ein, so meldet der Rechner :

```
  27  IST KEINE PRIMZAHL, SONDERN DURCH
  3   TEILBAR.

ENDE DER BERECHNUNGEN.
```

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 :

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

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

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

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

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

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 = ø ist; dann liegt aber auch eigentlich keine quadratische Gleichung mehr vor.

Deshalb betrachten wir im folgenden nur die "echten" Fälle ($a \neq ø$) und zugleich die im reellen Bereich lösbaren Fälle ($D \geq ø$). 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) :

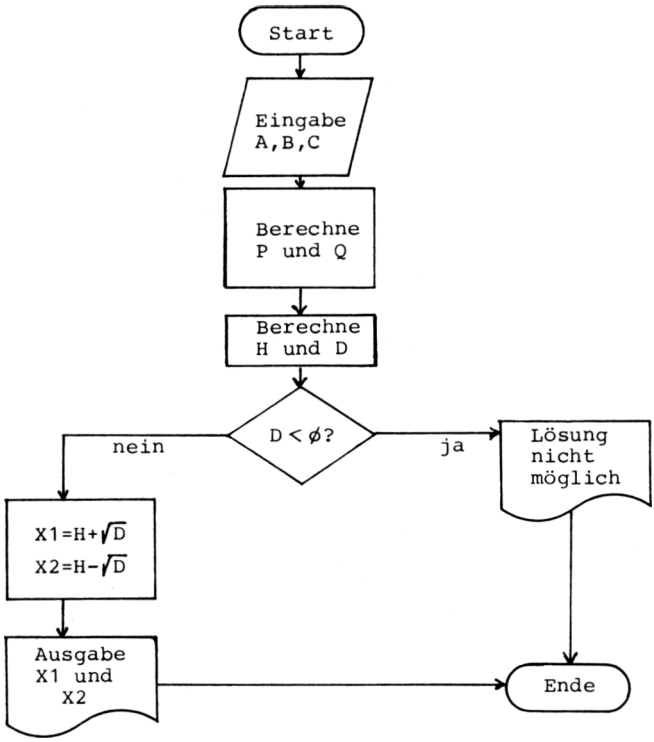
$$(3) \qquad x_{1/2} = -p/2 \pm \sqrt{p^2/4 - q}$$

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

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 $D \geq ø$ oder $D < ø$, die Lösungen bzw. eine Abbruchmeldung ausgeben.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM M4-QUADRATISCHE GLEICHUNG
20 CLS
30 PRINT"PROGRAMM ZUR LOESUNG EINER QUADRATISCHEN"
40 PRINT TAB(16)"GLEICHUNG."
50 PRINT:PRINT:PRINT TAB(10)"PROF.DR.W.VOSS, 1984"
: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"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

(6.) SCHRITT : Programmbeschreibung

Satz 1ø-5ø : Kommentar und Überschriftausgabe

Satz 6ø-8ø : Eingabe der Koeffizienten der quadratischen Gleichung (in der Form der Gleichung (1))

Satz 9ø : Bestimmung von P und Q gemäß Gleichung (2)

Satz 1øø : Bestimmung der Hilfsgröße H und der Diskriminate D

Satz 11ø : Wenn D kleiner als null ist, so erfolgt eine Abbruchmeldung auf dem Bildschirm und ein Sprung zu Satz 15ø

Satz 12ø : Ist die Diskriminante hingegen nicht negativ, so werden die beiden Lösungen X1 und X2 berechnet

Satz 13ø-14Ø: Beide Lösungen werden auf dem Bildschirm ausgegeben

Satz 15ø : Beendigung des Programms

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 :

$$e = (1 + 1/n)^n \qquad \text{für n gegen unendlich.}$$

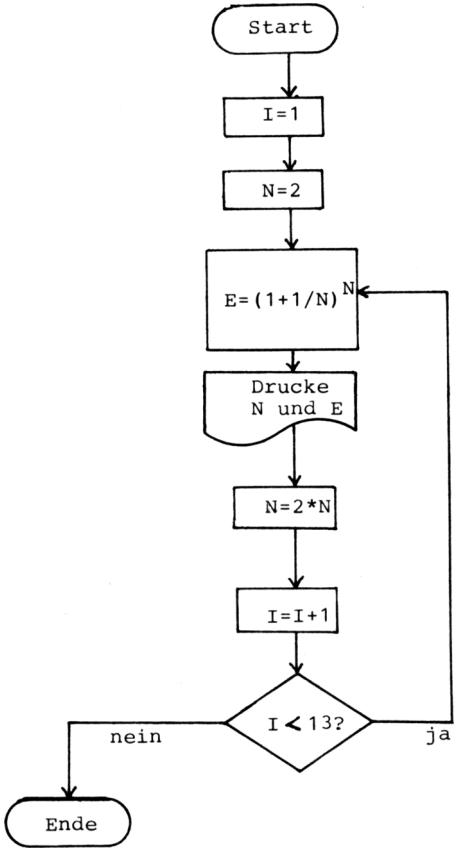
(2.) SCHRITT : Problemanalyse

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.

4. SCHRITT : Programm

```
10 REM M5-EULER
20 CLS
30 PRINT"PROGRAMM ZUR BESTIMMUNG DER ZAHL  E."
40 PRINT
50 PRINT:PRINT:PRINT TAB(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 PRINT N,E
110 N=N*2
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

6. SCHRITT : Programmbeschreibung

Satz 1ø-5ø : Überschrift

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

Satz 7ø : Ausgabe einer Tabellenüberschrift und einer Leerzeile

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

Satz 13ø : 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-

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

```
8ø FOR I = 1 TO 3ø
```

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

Die sich durch obiges Programm ergebende Tabelle sieht folgendermaßen aus :

```
    N            E

  2              2.25
  4              2.4414ø625
  8              2.56578451
  16             2.6379285ø
  32             2.67699ø13
  64             2.69734495
  128            2.7ø7739ø1
  256            2.71299162
  512            2.71563199
  1ø24           2.71695572
  2ø48           2.71761848
  4ø96           2.71795øø7
ENDE
```

2.7 Prozentrechnung

(1.) SCHRITT : Vorstellung des Problems

Beim Prozentrechnen spielt die Dreisätzrechnung 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 45ø.--. Der Kunde muß zusätzlich die sog. Mehrwertsteuer bezahlen, die derzeit 14 % des Nettorechnungsbetrages ausmacht. Somit ergibt sich also :

Bruttobetrag = Nettobetrag + 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.

(2.) SCHRITT : Problemanalyse

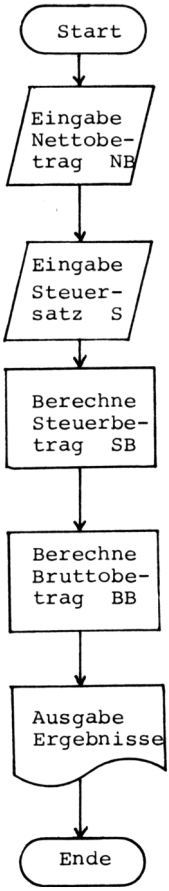
Das oben genannte Beispiel kann mit der Dreisatzrechnung folgendermaßen gelöst werden (und zeichnet uns so den Lösungsweg vor) :

Nettobetrag = 1øø % entsprechen 45ø.- DM
Steuerbetrag= 14 % entsprechen x DM

$$x = \frac{45ø * 14}{1øø} = 63.- \text{ (DM)}$$

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



4. SCHRITT : Programm

```
10 REM M6-PROZENT
20 CLS
30 PRINT"PROGRAMM ZUR BESTIMMUNG VON PROZENTWER-"
40 PRINT TAB(18)"TEN."
50 PRINT:PRINT:PRINT TAB(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"BRUTTOBETRAG  : ";BB
130 PRINT:PRINT"ENDE":END
```

5. SCHRITT : Variablenliste

BB = Bruttobetrag

NB = Nettobetrag

SB = Prozentbetrag

SS = Prozentsatz

6. SCHRITT : Programmbeschreibung

Satz 1ø-5ø : Überschrift.

Satz 6ø-7ø : Eingabe von Nettobetrag und Prozentsatz.

Satz 8ø-9ø : Berechnung von Prozentbetrag und Bruttobetrag.

Satz 95 : Zwei Leerzeilen.

Satz 1øø-12ø : Ausgabe der Ergebnisse auf dem Bildschirm.

Satz 13ø : Ende des Programms.

7. SCHRITT : Ergebnisse

Gibt man beispielsweise als Nettobetrag 2øø und als Prozentsatz den Wert 15 ein, so meldet das Programm :

```
NETTOBETRAG    :  2øø
PROZENTBETRAG  :  3ø
BRUTTOBETRAG   :  23ø
ENDE
```

28 Würfel

(1.) SCHRITT : Vorstellung des Problems

Es soll ein Programm vorgestellt werden, welches 1ØØ mal "würfelt" und den Durchschnitt dieser Augenzahlen ausgibt.

Man kann sich zu diesem Zweck der Funktion RND (RND = random = Zufall) bedienen, die Zufallszahlen zwischen Ø bis unter 1 erzeugt.

Zufallszahlen nennt man solche Zahlen, die in unregelmäßiger und unsystematischer Folge auftreten.

(2.) SCHRITT : Problemanalyse

Beim SCHNEIDER CPC 464 kann die Funktion RND ohne Argument benutzt werden.
Der Leser sollte diese Funktion einfach einmal erproben, indem er beispielsweise eingibt:

```
PRINT RND
```

und noch einmal:

```
PRINT RND
```

oder noch besser:

```
FOR I = 1 TO 1ø : PRINT RND : NEXT
```

und danach die ENTER-Taste.

Auf dem Bildschirm erscheinen Zufallszahlen, die alle zwischen ø bis unter 1 liegen.

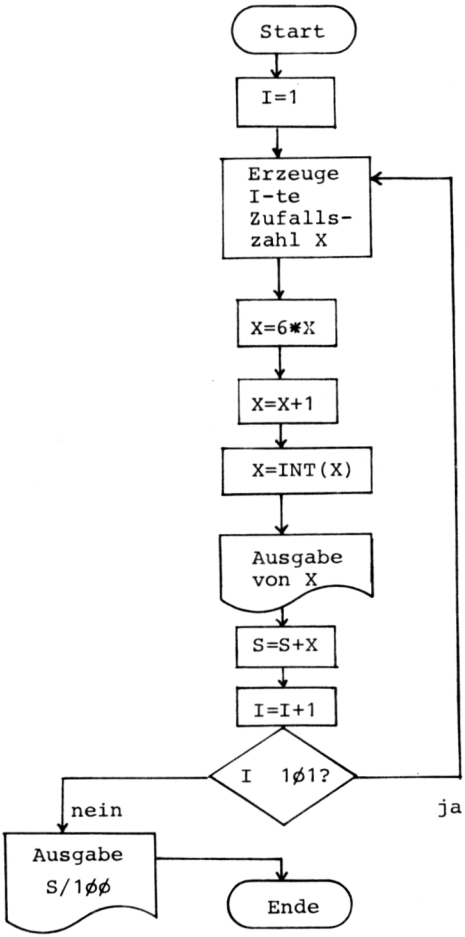
Wir hätten nun aber gern als Ergebnisse 1øø ganzzahlige Werte zwischen 1 und 6 - eben Würfel-Augenzahlen.

Das folgende Schema verdeutlicht, wie man aus Zufallszahlen aus dem Wertebereich zwischen ø bis unter 1 ganze Zahlen zwischen 1 und 6 erzeugen kann:

Verarbeitung	Ergebnis
RND-Funktion	zwischen ø bis unter 1
* 6	zwischen ø bis unter 6
+ 1	zwischen 1 bis unter 7
INT-Funktion	ganzzahlig zwischen 1 bis 6

Damit ist der Problemlösungsweg vorgezeichnet. Da schließlich noch der Durchschnitt gewünscht wird, sind alle Ergebnisse aufzuaddieren und die entstehende Summe durch 1øø zu teilen.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM M7-WUERFEL
20 CLS
30 PRINT"PROGRAMM ZUR ERZEUGUNG VON 100 WUERFEL"
40 PRINT"WUERFEN UND DER DURCHSCHNITTLICHEN AU-"
50 PRINT TAB(16)"GENZAHL."
60 PRINT:PRINT:PRINT
70 PRINT TAB(9)"PROF.DR.W.VOSS, 1984":PRINT
80 I=1
90 X=RND
100 X=X*6:X=X+1:X=INT(X)
110 PRINT X;
120 S=S+X
130 I=I+1
140 IF I<101 THEN 90
150 AM=S/100
160 PRINT:PRINT:PRINT
170 PRINT:PRINT"MITTELWERT = ";AM
180 PRINT:PRINT"ENDE DER AUSGABE":END
```

5. SCHRITT : Variablenliste

AM = Mittelwert
I = Laufindex (Rundenzähler)
S = Summe aller Würfelergebnisse
X = Zufallsergebnis

6. SCHRITT : Programmbeschreibung

Satz 1ø- 7ø : Überschrift

Satz 8ø : Vorgabe des Rundenzählers

Satz 9ø-1øø : Erzeugung einer Zufallszahl und Transformierung in einen ganzzahligen Wert zwischen 1 und 6

Satz 11ø : Ausgabe

Satz 12ø : Erhöhung der Summe S um das "Würfelergebnis" X

Satz 13ø : Erhöhung des Rundenzählers I um 1

Satz 14ø : Solange I kleiner als 1ø1 ist, geht es zurück zu Satz 9ø

Satz 15ø-17ø : Berechnung des Mittelwerts und Ausgabe

Satz 18ø : Beendigung des Programms

7. SCHRITT : Ergebnisse

Wenn wir dieses Programm starten, so erzeugt der Rechner nach der üblichen Überschrift 1øø zufällige Würfelergebnisse und gibt diese zeilenweise aus (der Leser überlege, warum eine zeilenweise Ausgabe erfolgt).
Danach wird der Durchschnitt aus den 1øø Ergebnissen berechnet und ausgegeben, der - wenn der Zufallszahlengenerator in Ordnung ist - in der Nähe von 3.5 liegen muß.

Da die Ergebnisse zufallsabhängig sind, können sie sinnvollerweise hier nicht dargestellt werden.

2.9 Probleme der Rechengenauigkeit

Ein Computer wie der SCHNEIDER CPC 464 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
 1.234567E+ø9
 1.234567E+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 :

1.234567E+ø9

Dies ist zu lesen als :

1.234567 * $1ø^{9}$ = 1.234567 * 1 øøø øøø øøø

Dieses Produkt wiederum ergibt :

1 234 567 ø̲ø̲ø̲

Der Rechner hat also beim Wert für C die letzten Stellen "unterschlagen"; er kann sie nicht mehr berücksichtigen.

Allerdings wird, wie man an entsprechenden Beispielen zeigen kann, bei diesem "Unterschlagen" nicht einfach

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.

Kapitel 3 : Chemie
==========

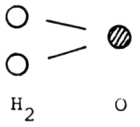
3.1 Zusätzliche BASIC-Anweisungen

Will man Programme aus dem Bereich der Chemie programmieren, so kann man hervorragend die Graphik- und Farbmöglichkeiten des SCHNEIDER CPC 464 ausnutzen - etwa zur graphischen Darstellung von Molekülen oder von chemischen Verbindungen.

Beispielsweise kann man das Wassermolekül

H_2O ,

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



H_2 O

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 spezieller Symbole (auf die Möglichkeiten der sog. hochauflösenden Graphik kommen wir erst in einem späteren Kapitel zu sprechen).

Zur Anfertigung solcher und ähnlicher Zeichnungen eignet sich hervorragend der Graphikzeichen-Vorrat des SCHNEIDER CPC 464 (siehe Bedienungshandbuch).

Diese Zeichen, die unter den ASCII-Codezahlen 126 ff zu finden sind (dabei sind auch noch einige spezielle Sonderzeichen und z.B. ein paar griechische Buchstaben) sind aufrufbar mit der früher kurz erwähnten Funktion

CHR$ (Argument)

Das Argument dieser Funktion ist die ASCII-Codezahl des gewünschten Symbols.

Gibt der Benutzer zum Beispiel ein:

```
PRINT  CHR$(231)
```

so erscheint auf dem Bildschirm ein Symbol, das einem Kügelchen ähnelt.

In diesem Zusammenhang einige Hinweise zur Farbgestaltung für die Glücklichen, die einen Farbfernseher angeschlossen haben (es sei darauf hingewiesen, daß diejenigen der folgenden Programme, die mit Farbeffekten arbeiten, auch bei angeschlossenen Schwarz-Weiß-Geräten funktionieren - unterschiedliche Farben stellen sich dann als unterschiedliche Grüntöne bzw. Schraffuren dar).

Wenn der SCHNEIDER CPC 464 eingeschaltet wird, befindet er sich im sog. Modus 1 (4o Zeichen pro Zeile) und bei angeschlossenem Farbmonitor werden die Ausgaben gelb auf dunkelblau dargestellt.

Die Ausgabefarben können mit folgender Anweisung geändert werden:

Statement 12:

```
nn  PEN  Zahl
```

Im Modus 1 sind mit dieser Anweisung vier unterschiedliche Farben gemäß der folgenden Angaben darstellbar:

Zahl	Farbe
1	gelb
2	hellblau
3	rot
4	blau

Die letzte Farbe führt allerdings zu Ausgaben, die auf dem blauen Hintergrund nicht sichtbar werden.

Mit dem folgenden Programm können drei verschiedenfarbige Kügelchen untereinander ausgegeben werden:

```
10 FOR I=1 TO 3
20 PEN 1
30 PRINT I,CHR$(231)
40 NEXT 1
50 END
```

Hinweis:

Wenn beim Farbexperimentieren (oder auf andere Weise) der Benutzer in eine "Sackgasse" gerät, in der "nichts mehr geht", so kann er den Rechner in den Grundzustand versetzen durch gleichzeitiges Drücken der Tasten

CTRL SHIFT und ESC

Allerdings geht dann ein gerade im Rechner befindliches Programm verloren.

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 Rechner zu bekommen.

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

Statement 13:

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

Statement 14:

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

Beispiel :

```
1ø REM DATENEINGABE
2ø DATA 46øø,DORTMUND,463ø,BOCHUM,8øøø
3ø DATA MUENCHEN,2øøø,HAMBURG
4ø READ P1,O1$,P2,O2$,P3,O3$,P4,O4$
5ø ?P1,O1$:?P2,O2$:?P3,O3$:?P4,O4$
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, ... allgemein x_i

Dies geht auch in entsprechender Weise in der Programmiersprache BASIC :

X(1), X(2), X(3), ...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 15:

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

Beispielsweise hält das Statement

```
1ø DIM X(2ø)
```

für die Variable X 21 Speicherplätze frei.

Es handelt sich dabei um die Plätze X(ø), X(1), X(2), X(3), ... , X(2ø).

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

Das Statement

```
1ø  DIM Z(2,3)
```

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

Z(ø,ø)	Z(ø,1)	Z(ø,2)	Z(ø,3)
Z(1,ø)	Z(1,1)	Z(1,2)	Z(1,3)
Z(2,ø)	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 16:

```
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.

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 gelbe 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 :

1. Zeile : Ein gelbes Kügelchen und ein Schrägstrich von links oben nach rechts unten.

2. Zeile : Ein rotes Kügelchen (nach rechts versetzt).

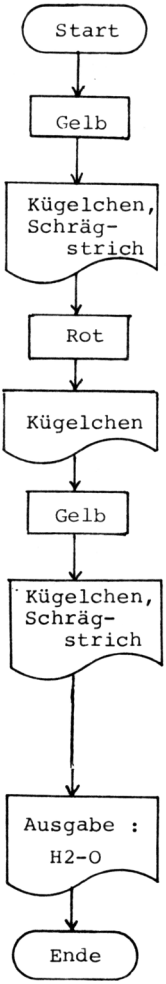
3. Zeile : Ein gelbes Kügelchen unter dem der ersten Zeile und ein Schrägstrich von links unten nach rechts oben.

4. Zeile : Text : H2-O .

Das Kügelchen erzeugen wir, wie schon zuvor, mit dem ASCII-Codezeichen mit der Nummer 231 (mit Hilfe der CHR$-Funktion).

Entsprechend hat der Schrägstrich von links oben nach rechts unten (\) die Codezahl 2ø5 und der umgekehrte Schrägstrich (/) die Codezahl 2ø4 (zu entnehmen dem Anhang III der englischen Version des Betriebshandbuchs).

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM C1-H2O
20 CLS
30 PRINT"PROGRAMM ZUR GRAPHISCHEN DARSTELLUNG"
40 PRINT"          DES WASSERMOLEKUELS H2-O"
50 PRINT:PRINT:PRINT TAB(10)"PROF.DR.W.VOSS, 1984"
:PRINT
60 PRINT:PRINT:PRINT
70 PEN 1:PRINT TAB(10)CHR$(231);
80 PRINT TAB(12)CHR$(205)
90 PEN 3:PRINT TAB(14)CHR$(231)
100 PEN 1:PRINT TAB(10)CHR$(231);
110 PRINT TAB(12)CHR$(204)
130 PRINT:PRINT TAB(10)"H2 -O"
140 PRINT:PRINT:PRINT"ENDE":END
```

5. SCHRITT : Variablenliste

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

6. SCHRITT : Programmbeschreibung

Satz 1ø-6ø : Überschrift

Satz 7ø : Umschalten auf Gelb und Ausgabe eines Kügelchens in Spalte 1ø; Unterdrückung des Zeilenvorschubs

Satz 8ø : Ausgabe eines Schrägstrichs in Spalte 12

Satz 9ø : Umschalten auf Rot und Ausgabe eines Kügelchens in Spalte 14

Satz 1øø : Umschalten auf Gelb und Ausgabe eines Kügelchens in Spalte 1ø; Unterdrückung des Zeilenvorschubs

Satz 11ø : Ausgabe eines Schrägstrichs in Spalte 12

Satz 13ø : Ausgabe von H2- O ab Spalte 1ø nach zwei Leerzeilen

Satz 14ø : 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 :

$$2\,C + O_2 \longrightarrow 2\,CO$$

Koks	Sauerstoff	Kohlenmonoxid

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

$$Fe_2O_3 + 3\ CO \longrightarrow 2\ Fe + 3\ CO_2$$

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.

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 CLS
30 PRINT"PROGRAMM ZUR DARSTELLUNG EINER CHEMI-"
40 PRINT TAB(13)"SCHEN REAKTION."
50 PRINT:PRINT:PRINT TAB(10)"PROF.DR.W.VOSS. 1984"
:PRINT
60 PRINT:PRINT:PRINT
70 PEN 1:PRINT TAB(10)CHR$(231):
75 PRINT TAB(25)CHR$(231);:PEN 2:PRINT CHR$(231)
80 PRINT:PRINT TAB(12)"+   ";:PEN 2:PRINT CHR$(231)
;:PRINT CHR$(231);
85 PRINT"    =    "
86 PRINT
90 PEN 1:PRINT TAB(10)CHR$(231):
100 PRINT TAB(25)CHR$(231);:PEN 2:PRINT CHR$(231)
110 PRINT:PRINT
120 PEN 1:PRINT TAB(10)"2C + O     =     2CO"
130 PRINT TAB(16)"2"
140 PRINT:PRINT:PRINT:PRINT"BITTE CONT EINGEBEN":S
TOP
```

```
150 CLS
160 PEN 2:PRINT TAB(33)CHR$(231)
170 PEN 1:PRINT TAB(18)CHR$(231);:PEN 2:PRINT CHR$
(231);
180 PEN 1:PRINT TAB(32)CHR$(231)
190 PEN 2:PRINT TAB(12)CHR$(231);:PEN 3:PRINT TAB(
26)CHR$(231);:PEN 2:PRINT TAB(33)CHR$(231)
200 PEN 3:PRINT TAB(10)CHR$(231);:PEN 2:PRINT TAB(
33)CHR$(231)
210 PRINT TAB(12)CHR$(231);:PRINT TAB(15)"+";
220 PEN 1:PRINT TAB(18)CHR$(231);:PEN 2:PRINT TAB(
19)CHR$(231);
230 PEN 1:PRINT TAB(23)"=";:PRINT TAB(29)"+";:PEN
1:PRINT TAB(32)CHR$(231)
240 PEN 3:PRINT TAB(10)CHR$(231);:PEN 2:PRINT TAB(
33)CHR$(231)
250 PRINT TAB(12)CHR$(231);:PEN 3:PRINT TAB(26)CHR
$(231);:PEN 2:PRINT TAB(33)CHR$(231)
260 PEN 1:PRINT TAB(18)CHR$(231);:PEN 2:PRINT TAB(
19)CHR$(231);
270 PEN 1:PRINT TAB(32)CHR$(231)
280 PEN 2:PRINT TAB(33)CHR$(231)
290 PRINT:PRINT
300 PEN 1:PRINT TAB(8)"FE O   +  3CO  =  2FE + CO"
310 PRINT TAB(10)"2 3                     2"
320 PRINT:PRINT"ENDE DER AUSGABE":END
```

5. SCHRITT : Variablenliste

Es werden keine Variablen benutzt.

6. SCHRITT : Programmbeschreibung

Satz 1ø- 6ø	: Überschrift
Satz 7ø	: Ausgabe eines gelben Kügelchens in Spalte 1ø
Satz 75	: Ausgabe eines gelben Kügelchens in Spalte 25 und daneben eines blauen Kügelchens
Satz 8ø	: Ausgabe eines +-Zeichens in Spalte 12 und daneben zweier blauer Kügelchen
Satz 85	: Ausgabe eines =-Zeichens in der gleichen Zeile
Satz 86	: Leerzeile
Satz 9ø	: wie Satz 7ø
Satz 1øø	: wie Satz 75
Satz 11ø-13ø	: Ausgabe der Reaktionsgleichung
Satz 14ø-15ø	: Abwarten und Räumen des Bildschirms

Satz 16ø-28ø : Ausgabe des graphischen Abbilds der zweiten Reaktionsgleichung

Satz 29ø-31ø : Ausgabe der jeweiligen Reaktionsgleichung

Satz 32ø : Ende des Programms

(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 1øø Gramm Zink vollständig zu Zinksulfid umzusetzen ? (Schwefel = S; Zink = Zn; Zinksulfid = ZnS)

Atomgewicht von Schwefel : 32.1
Atomgewicht von Zink : 65.4

(2.) SCHRITT : Problemanalyse

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

$$Zn + S \longrightarrow ZnS$$

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 :

$$x/1\emptyset\emptyset = 32.1/65.4$$

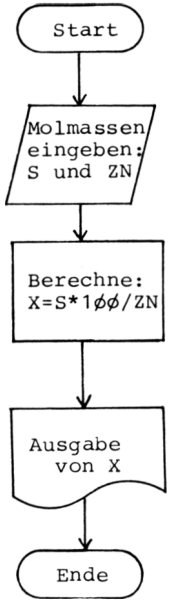
oder

$$x = 321\emptyset/65.4$$

Dabei ist x der gesuchte Wert.

(3.) SCHRITT : Flußdiagramm

Das Flußdiagramm ist bei dieser Fragestellung wieder sehr einfach :



4. SCHRITT : Programm

```
10 REM C3-CHEM.RECHNEN
20 CLS
30 PRINT"PROGRAMM ZUM STOECHIOMETRISCHEN RECHNEN"
40 PRINT:PRINT
50 PRINT:PRINT:PRINT TAB(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:PR
INT
90 PRINT:PRINT"WIEVIEL GRAMM SCHWEFEL SIND NOTWEND
IG,"
100 PRINT"UM 100 G ZINK VOLLSTAENDIG 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$;:PRINT TAB(30)":";:
INPUT A
160 PRINT"ATOMGEWICHT VON ";B$;:PRINT TAB(30)":";:
INPUT B
170 X=B*100/A
180 PRINT:PRINT:PRINT"BENOETIGTE MENGE VON ";B$;"
: ":PRINT:PRINT TAB(5)X;" G"
190 PRINT:PRINT:PRINT"ENDE DER BERECHNUNG":END
```

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 1ø-11ø : Überschrift und Beispielsangabe mit der in diesem Programm zu behandelnden Fragestellung

Satz 12ø-13ø : Eingabe der Namen der beiden Stoffe

Satz 14ø : 3 Leerzeilen

Satz 15ø-16ø : Ausgabe der Namen der Stoffe und Anforderung der Atomgewichte dieser Stoffe

Satz 17ø : Berechnung der gesuchten Menge

Satz 18ø : Ausgabe des Ergebnisses (zuzüglich einiger Leerzeilen, auf die aber zukünftig nicht immer gesondert aufmerksam gemacht werden soll)

Satz 19ø : Ende des Programms

(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 :

    2ø3.738318 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 :

1. Chemisches Zeichen,
2. Ordungszahl im Periodensystem,
3. Atomgewicht,
4. Wichte,
5. Schmelzpunkt,
6. Siedepunkt,
7. Gruppenzugehörigkeit (I-VIII,ø),
8. Nebengruppenzugehörigkeit (a,b)
9. Elektronenschalen (K-Q)

(2.) SCHRITT : Problemanalyse

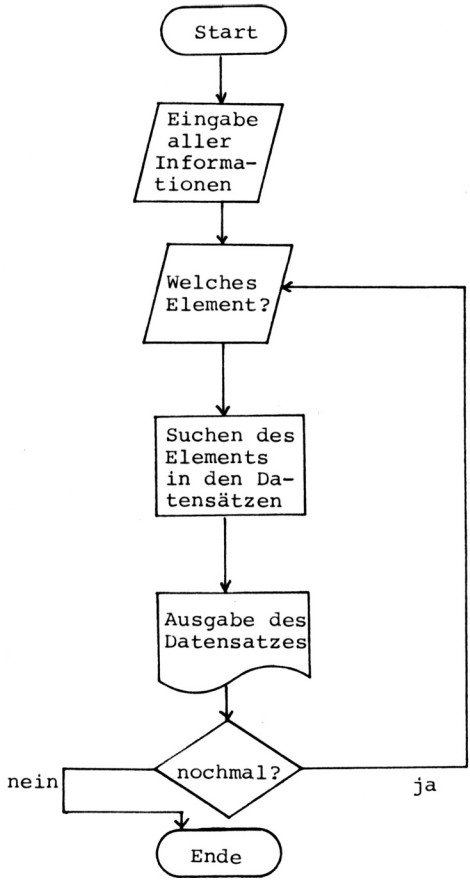
Die Entwicklung eines entsprechenden Computerprogramms, das die interessierenden Auskünfte bereitstellen kann, bereitet keine besonderen Schwierigkeiten.

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.

Schließlich kann abgefragt werden, ob für ein weiteres chemisches Element diese Informationen gewünscht werden.

3. SCHRITT : Flußdiagramm

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



4. SCHRITT : Programm

```
10 REM C4-ELEMENTE
20 CLS
30 PRINT"PROGRAMM ZUR ANGABE WESENTLICHER EIGEN-"
40 PRINT TAB(5)"SCHAFTEN CHEMISCHER ELEMENTE."
50 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984":
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 NEXT J
100 FOR I=1 TO N
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=1 TO N: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 CLS
170 PRINT N$(I,1);:PRINT TAB(23)N$(I,2)
180 PRINT: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 CLS : GOTO 110
310 PRINT:PRINT"ENDE IN 310":END
```

```
495 REM DATEN BEGRIFFE
500 DATA ALUMINIUM,ANTIMON,ARGON,ARSEN,BARIUM,BLEI
,BOR,BROM,CHLOR,CHROM
510 DATA EISEN,FLUOR,GOLD,HELIUM,IRIDIUM,JOD,KADMI
UM,KALIUM,KALZIUM,KOBALT
520 DATA KOHLENSTOFF,KUPFER,LITHIUM,MAGNESIUM,MANG
AN,MOLYBDAEN,NATRIUM
530 DATA NEON,NICKEL,PHOSPHOR,PLATIN,PLUTONIUM,QUE
CKSILBER,RADIUM,RADON
540 DATA SAUERSTOFF,SCHWEFEL,SELEN,SILBER,SILIZIUM
,STICKSTOFF
550 DATA STRONTIUM,THORIUM,TITAN,URAN,VANADIUM,WAS
SERSTOFF,WISMUT,WOLFRAM
570 DATA XENON,ZINK,ZINN
595 REM DATEN ABKUERZUNGEN
600 DATA AL,SB,AR,AS,BA,PB,B,BR,CL,CR,FE,F,AU,HE,I
R,J,CD,K,CA,CO,C,CU,LI,MG
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
620 DATA BI,W,XE,ZN,SN
640 REM DATEN ORDNUNGSZAHLEN
650 DATA 13,51,18,33,56,82,5,35,17,24,26,9,79,2,77
,53,48,19,20,27,6,29
660 DATA 3,12,25,42,11,10,28,15,78,94,80,88,86,8,1
6,34,47,14,7,38
670 DATA 90,22,92,23,1,83,74,54,30,50
695 REM DATEN ATOMGEWICHTE
700 DATA 26.97,121.76,39.94,74.91,137.36,207.21,10
.82,79.92,35.46,52.01
710 DATA 55.85,19.197,4.003,192.2,126.92,112.41,39
.096,40.08,58.94,12.011
720 DATA 63.54,6.94,24.32,54.94,95.95,22.991,20.18
3,58.71,30.98,195.09
730 DATA 242,200.61,226.05,222,16,32.066,78.96,107
.88,28.06,14.008
740 DATA 87.63,232.05,47.9,238.07,50.95,1.008,209,
183.86,131.3,65.38,118.7
745 REM DATEN WICHTE
750 DATA 2.7,6.67,.0018,5.72,3.5,11.34,1.73,3.14,1
.557,7.14,7.86,.0017
760 DATA 19.25,.00018,22.42,4.942,8.64,.862,1.545,
8.83,3.51,8.933,.534
770 DATA 1.75,7.3,10.2,.971,.0009,8.8,1.83,21.4,-,
13.55,-,.009,.0014,2.07
780 DATA 4.46,10.5,2.34,.00125,2.63,11.7,4.43,19,6
.07,.0009,9.8,19.3
790 DATA .0059,7.14,7.28
795 REM DATEN SCHMELZPUNKTE
800 DATA 658,630,-190,,704,327,2400,-7.3,-100,1800
,1525,-218,1063,-272
810 DATA 2454,114,321,64,851,1490,-,1083,180,650,1
221,2622,99
820 DATA -248.6,1452,44,1771,-,-39,700,-71,-218.7,
113,220,960,1414
830 DATA -210,771,1827,1727,1689,1726,-262,271,338
0,-111.9,419,232
```

```
845 REM DATEN SIEDEPUNKT
850 DATA 2270,1440,-186,616,1537,1690,2550,58,-34,
2600,2450,-187,2677
860 DATA -269,4400,184,770,762,1439,3185,-,2360,13
72,1120,2152,4800,880
870 DATA -246,3075,280,3800,-,357,1140,-61.9,-182.
97,444,688,1930,2630
880 DATA -195.8,1360,3530,-,-,3000,-272.78,1560,60
00,-108.1,908,2430
900 REM DATEN KLASSEN
910 DATA III,B,M,V,B,O,O,A,M,V,B,N,II,A,P,IV,B,P,I
II,B,L,VII,B,N
920 DATA VII,B,M,VI,A,N,VIII,A,N,VII,B,L,I,B,P,O,A
,K,VIII,A,P,VII,B,O
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
940 DATA VI,A,O,I,A,M,O,A,L,VIII,A,N,V,B,M,VIII,A,
P,-,-,-,II,B,P,II,A,O,O,A,P
950 DATA VI,B,L,VI,B,M,VI,B,N,I,B,O,IV,B,M,V,B,L,I
I,A,O,-,-,-,IV,A,N
960 DATA -,-,-,V,A,N,I,A,K,V,B,P,VI,A,P,O,A,O,II,B
,N,IV,B,O
```

(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 1ø-5ø : Überschrift

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

Satz 6ø : Angabe der Elementanzahl (es sind in dieses Programm nicht alle chemischen Elemente aufgenommen, sondern nur 52 Stück. Ggf. muß also Satz 6ø verändert werden, falls die DATA-Statements verlängert werden sollten)

Satz 7ø : Dimensionierung

Satz 75-1ø8 : Einlesen aller Informationen

Satz 11ø : Anforderung des Elements, über das Informationen gewünscht werden

Satz 12ø-13ø : Aufsuchen des gewünschten Elements im Datenbestand;

wird es gefunden, weiter bei 16ø;
wird es hingegen nicht gefunden, weiter bei 14ø

Satz 14ø-15ø : Ist das gewünschte Element nicht im Datenbestand vorhanden, so erfolgt ein entsprechender Ausdruck; danach Sprung zum Satz 29ø

Satz 16ø : Löschen des Bildschirms

Satz 17ø : Ausgabe des Elementnamens und des chemischen Kürzels

Satz 18ø : Leerzeile

Satz 19ø-28ø : Ausgabe der übrigen Informationen für das gewünschte Element

satz 29ø-3øø : Abfrage, ob eine weitere Auskunft gegeben werden soll;

wenn ja, Löschen des Bildschirms und

Sprung nach 11ø

wenn nein, weiter bei 31ø

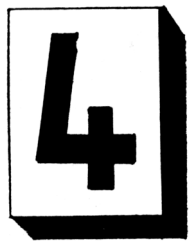
Satz 31ø : Beendigung des Programms

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)  ?
```



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

Verwendet man den SCHNEIDER CPC 464 zur Erstellung von Graphiken, so steht die sog. hochauflösende Graphik im Mittelpunkt des Interesses.

Je nachdem, in welchem Bildschirmmodus der Rechner arbeitet, steht ein unterschiedlich hohes Auflösungsvermögen (und - wie schon erwähnt wurde - unterschiedlich breite Farbpaletten) zur Verfügung.

Mit dem folgenden Kommando wird der Bildschirmmodus festgelegt:

Kommando 8 :

```
MODE  Zahl
```

An der Stelle "Zahl" kann dabei eine Null, Eins oder Zwei stehen.

Diese Zahlenangaben haben die folgende Bedeutung:

MODE	Zeichen pro Zeile
ø	2ø
1	4ø
2	8ø

Nach dem Einschalten befindet sich der Rechner per Voreinstellung im Modus 1, stellt also 4ø Zeichen pro Bildschirmzeile zur Verfügung.

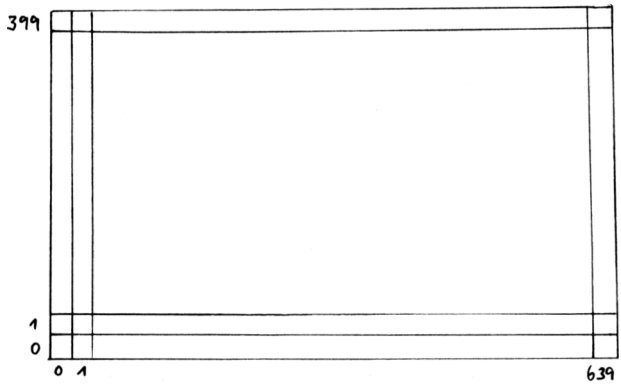
Generell bietet er für die hochauflösende Graphik ein Auflösungsvermögen von 64ø * 4øø Punkten (64ø Spalten und 4øø Zeilen).

Dieses Auflösungsvermögen macht es möglich, relativ feine Linienzüge auf dem Bildschirm zu erzeugen, wenn man die dafür notwendigen Anweisungen kennt.

Die wichtigsten dieser Anweisungen sollen im folgenden kurz besprochen werden und an einer Reihe einfacher und überschaubarer Beispiele erprobt werden.

Wenn man hochauflösende Graphiken anfertigt, ist es sinnvoll, sich vorzustellen, daß auf dem Bildschirm sich nun ein (nicht sichtbarer) Graphik-Cursor bewegt, dessen Position angibt, an welcher Stelle gerade gezeichnet werden kann.

Wir sagten schon, daß man sich im Graphikmodus den Bildschirm gemäß der folgenden Skizze aufgeteilt denken muß:



Besonders wichtig ist in diesem Zusammenhang, daß die Zeile ø die unterste Zeile, die Spalte ø die am weitesten links befindliche Spalte des Bildschirms ist.

Somit befindet sich der Koordinatenpunkt (ø,ø), also der Ursprung eines Achsenkreuzes, das man sich denken kann, links unten - so wie man es beispielsweise aus der Schulmathematik auch gewohnt ist.

Will man nun auf dem Bildschirm einen Punkt zeichnen, so dient dazu die folgende Anweisung:

Statement 17:

```
PLOT  S,Z
```

Dabei steht S für die Spalte (ø-639) und Z für die Zeile (ø-399), in die gezeichnet werden soll.

Der Leser erprobe das folgende Programm:

```
1ø  CLS
2ø  PLOT 32ø,2øø
3ø  END
```

Das Ergebnis ist ein kleiner Punkt auf der Mitte des Bildschirms.

Entsprechend einfach ist das Ziehen von geraden Linien mit dem folgenden Statement:

Statement 18:

```
DRAW  S,Z
```

4.3 BASIC-Programme

So zieht beispielsweise das Statement

DRAW 1ØØ,1ØØ

einen geraden Strich von der jeweiligen Position des Graphik-Cursors ausgehend bis zum Koordinatenpunkt (1ØØ,1ØØ).

Wenn der Graphik-Cursor durch ein vorhergehendes PLOT- oder DRAW-Statement nicht bewegt wurde, wird dieser Strich von (Ø,Ø) nach (1ØØ,1ØØ) gezogen, also von links unten ca. 1Ø cm schräg aufwärts.

Nehmen wir an, wir wollen in die Mitte des Bildschirms einen senkrechten Strich ziehen.
Dazu benötigen wir das folgende Programm:

```
10 CLS
20 PLOT 320,0
30 DRAW 320,399
40 END
```

Entsprechend sieht ein Programm für einen waagrechten Strich in der Mitte des Bildschirms folgendermaßen aus:

```
10 CLS
20 PLOT 0,200
30 DRAW 639,200
40 END
```

Entsprechend einfach ist es, beispielsweise ein Achsenkreuz zu zeichnen, wie das folgende Programm illustriert:

```
10 CLS
20 PLOT 0,0
30 DRAW 400,0
40 PLOT 0,0
50 DRAW 0,250
60 END
```

Sehr leicht ist es bei diesem Rechner - im Gegensatz zu vielen anderen - in eine graphische Darstellung Texte einzufügen.

Natürlich können auch Graphiken unter Benutzung spezieller Sonderzeichen erzeugt werden, die der ASCII-Code für den SCHNEIDER CPC 464 bereitstellt.

Sie finden sich im Bereich der Codeziffern 123 bis 255 und können aufgerufen werden durch die Anweisung

PRINT CHR$ (Zahl)

für Zahl = 123 bis 255

Der Leser erprobe das folgende Programm, welches alle Sonderzeichen auf dem Bildschirm ausgibt. Es bleibt allerdings nach je 15 Zeichen stehen, damit der Benutzer sich diese Symbole in Ruhe anschauen kann und muß dann durch Eintippen des Kommandos

CONT

wieder in Gang gesetzt werden.

```
10 CLS
20 PRINT"NR.  SYMBOL":PRINT
30 PRINT
40 FOR I= 123 TO 255
50 PRINT I;"      ";CHR$(I)
60 IF I/15=INT(I/15) THEN STOP
70 NEXT I
80 END
```

Allerdings kann auch mit den Sonderzeichen "gezeichnet" werden, die über die CHR$-Funktion bereitgestellt werden können.

Beispielsweise finden wir unter der Codezahl 124 einen senkrechten Strich von ca. 5 mm Länge.
Wenn wir nun programmieren:

```
10 CLS
20 PRINT:PRINT:PRINT
30 FOR I=1 TO 10
40 PRINT TAB(10)CHR$(124)
50 NEXT I
60 END
```

so erhalten wir einen über 1ø Zeilen reichenden senkrechten Strich in Spalte 1ø des Bildschirms, wobei dieser sich dabei wieder im Normalgraphikmodus befindet, also 4ø Spalten zu je 25 Zeilen aufweist.

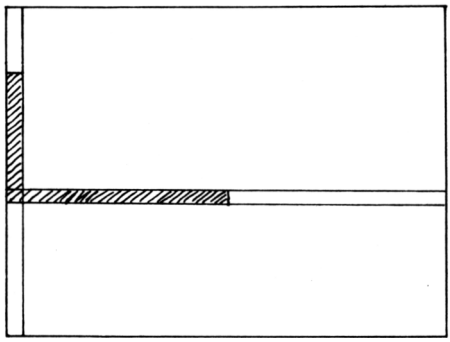
Entsprechend kann auch mit allen anderen Symbolen unter Benutzung der Funktion CHR$ umgegangen werden.

Mit diesen Symbolen kann auch gezeichnet werden, wobei aber im Modus 1 auf dem Bildschirm nur 4ø Spalten und 25 Zeilen zur Verfügung stehen.

Es handelt sich also nicht um eine hochauflösende Graphik, wie das folgende Beispiel zeigt:

```
10 CLS
20 PRINT:PRINT:PRINT:PRINT
30 FOR I=1 TO 10
40 PRINT CHR$(143)
50 NEXT I
60 FOR I=1 TO 20
70 PRINT CHR$(143);
80 NEXT I
90 PRINT:PRINT:PRINT"ENDE":END
```

Dieses Programm erzeugt schematisch das folgende Bild:



Natürlich kann man bei derartigen Graphiken die Farbe wechseln.
Wenn wir z.B. in das obige Programm die Sätze

```
15  PEN 2
```

und

```
85  PEN 1
```

einfügen, dann wird das dicke Achsenkreuz in himmelblau und der Schlußausdruck "ENDE" wieder in gelb ausgegeben.

Bei der Durchsicht des vorangegangenen Programms wird der Leser bemerkt haben, daß die Verschiebung des Balkenkreuzes nach unten dadurch erreicht wurde, daß das Programm mit einer Reihe von PRINT-Statements eingeleitet wurde, um Leerzeilen zu erzeugen.

Derartige Verschiebungen bzw. Plazierungen an beliebiger Stelle des Bildschirms können einfacher mit dem folgenden Statement erreicht werden:

Statement 19:

```
nn  LOCATE  S,Z
```

Dieses Statement rückt den Cursor an die Bildschirmstelle, die durch Spalte S (1-4Ø) und Zeile Z (1-25) gekennzeichnet wird.

Wollen wir beispielsweise das Wort "AHA" in die Mitte des Bildschirms schreiben, so gelingt dies mit folgendem Programm:

```
10 CLS
20 LOCATE 19,12
30 PRINT"AHA"
40 END
```

Mit diesem LOCATE-Statement ist man nun auch in der Lage, in sehr einfacher Weise Graphiken mit Texten zu mischen, was bei anderen Rechnern - zumindest bei der hochauflösenden Graphik - häufig mit großen Problemen verbunden ist.

Beispielsweise will man die Achsen eines Koordinatensystems mit den Bezeichnungen "EINKOMMEN" (waagrechte Achse) und "ANZAHL" (senkrechte Achse) beschriften.

Dies leistet das folgende Programm:

```
10 CLS
20 LOCATE 1,10
30 PRINT"ANZAHL"
40 LOCATE 25,22
50 PRINT"EINKOMMEN"
60 PLOT 50,50
70 DRAW 350,50
80 PLOT 50,50
90 DRAW 50,220
100 END
```

Es leuchtet ein, daß nach diesem Muster auch entsprechend andere Aufgaben gelöst werden können.

Nach diesen ausführlichen generellen Anmerkungen zur Graphikprogrammierung werden wir uns nun wieder konkreten Problemstellungen zu.

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.

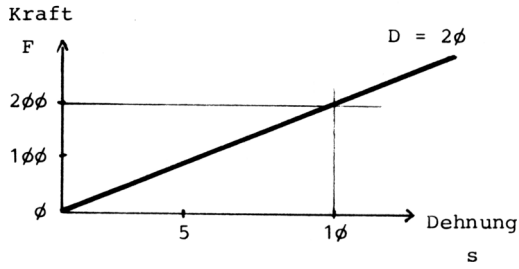
Es gibt also eine sog. Federkonstante, nämlich den Quotienten aus Gewichtskraft F und Federverlängerung s, d.h. es gilt :

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

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

(2.) SCHRITT : Problemanalyse

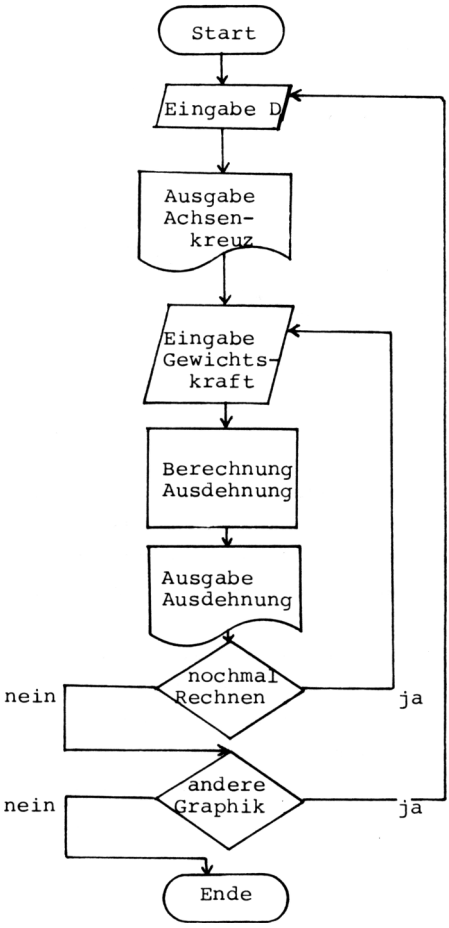
Der oben geschilderte Zusammenhang läßt sich (z.B. für D = 2ø) folgendermaßen graphisch darstellen :



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



4. SCHRITT : Programm

```
10 REM P1-FEDERWAAGE
20 CLS
30 PRINT"PROGRAMM ZUR GRAPHISCHEN VERANSCHAULI-"
40 PRINT TAB(3)"CHUNG DES FEDERWAAGEN-GESETZES."
50 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984":
PRINT
55 PRINT:PRINT:PRINT:PRINT
60 INPUT "WERT DER FEDERKONSTANTEN D : ";D
70 CLS:REM ACHSENKREUZ
75 LOCATE 3,3:PRINT"KRAFT"
77 LOCATE 20,3:PRINT" D = ";D
80 PLOT 50,50:DRAW 380,50
90 PLOT 50,50:DRAW 50,350
100 LOCATE 25,22:PRINT"CM"
110 PLOT 50,50
120 B=D/10
125 A=380
130 E=50+B*350
140 IF E>399 THEN A=400-(350*(E-399)/(E-49)):E=399
150 DRAW A,E
263 FOR I= 1 TO 9999:NEXT I
265 CLS
270 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 CLS:GOTO 60
350 PRINT:PRINT:PRINT"ENDE":END
```

(5.) SCHRITT : Variablenliste

A = Spalte für das Ende der Konstanzlinie

A$ = Antwortstring (Ja/Nein)

B = D/1ø (Größe zur Normierung der Bildschirmausgabe)

D = Federkonstante

E = Zeile für das Ende der Konstanzlinie

F = Kraft

FW = Federweg

I = Laufindex

(6.) SCHRITT : Programmbeschreibung

Satz 1ø- 55 : Überschrift

Satz 6ø : Eingabe der Federkonstanten D

Satz 7ø-1øø : Zeichnen des Achsenkreuzes

Ursprung: 5ø,5ø
waagrechte Achse: 5ø,5ø bis 38ø,5ø
senkrechte Achse: 5ø,5ø bis 5ø,35ø

Angabe der Achsenbezeichnungen (Satz 75 und Satz 1øø) und Angabe der Funktionsbezeichnung (Satz 77)

Satz 11ø-15ø : Zeichnen der Konstanzlinie von 5ø,5ø bis A,E

Dabei müssen der ursprüngliche Wert für A(A=38ø; Satz 125) und der Wert für E verändert werden, falls E größer als 399 wird, gemäß folgender Skizze

Satz 263-265 : Warteschleife und Löschen des Bildschirms

Satz 27ø : Anforderung einer Kraft F

Satz 28ø : Berechnung des Federweges

Satz 29ø-3øø : Ausgabe der Ergebnisse

Satz 31ø : Abfrage, ob noch eine Rechnung;

wenn ja, zurück nach 265
wenn nein, weiter bei 33ø

Satz 33ø : Abfrage, ob eine andere Graphik

wenn ja, zurück nach Satz 6ø nach Löschen des Bildschirms

wenn nein, weiter bei Satz 35ø

Satz 35ø : 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).

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 unbekannten 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.

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 :

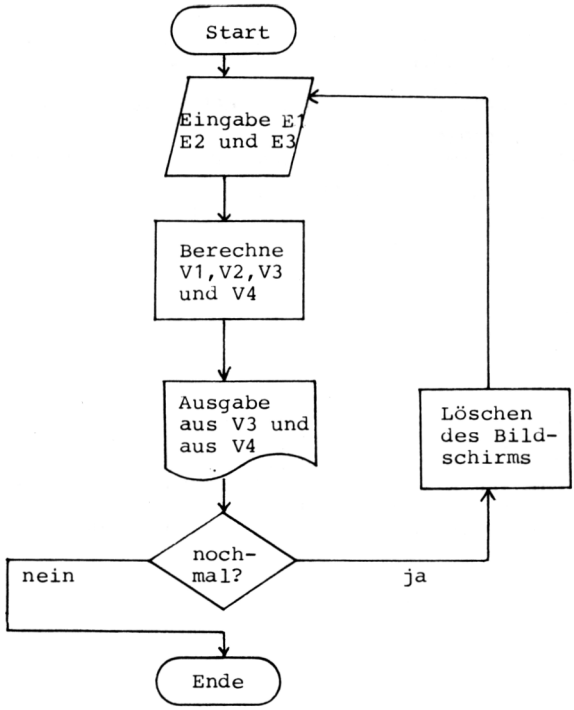
E1. Gewicht des unbekannten Körpers in der Luft;
E2. Scheinbares Gewicht des Körpers in der Flüssigkeit, in die er eingetaucht wird;
E3. Wichte der Flüssigkeit, die zur Überprüfung verwendet wird.

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 V3 und V4 erzielten Ergebnisse vorzusehen.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM P2-ARCHIMEDES
20 CLS
30 PRINT"PROGRAMM ZUR VERWENDUNG DES SATZES"
40 PRINT TAB(12)"DES ARCHIMEDES.":PRINT:PRINT
50 PRINT TAB(9)"PROF.DR.W.VOSS, 1984":PRINT
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 PRINT TAB(5)"- GEWICHT DES KOERPERS AN DER LUFT
"
100 PRINT TAB(5)"  (GL)"
110 PRINT:PRINT TAB(5)"- SCHEINBARES GEWICHT DES K
OERPERS"
120 PRINT TAB(5)"  IN DER FLUESSIGKEIT  (GS)"
130 PRINT:PRINT TAB(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 CLS
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 CLS:GOTO 70
260 PRINT:PRINT"ENDE DER BERECHNUNGEN":END
```

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 1ø-13ø : Überschrift und erläuternde Informationen

Satz 14ø-16ø : Anforderung der Input-Informationen

Satz 17ø-19ø : Verarbeitungsschritte (Berechnungen)

Satz 2øø-23ø : Ausgabe der Ergebnisse

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

(7.) SCHRITT : Ergebnisse

Geben wir zum Beispiel die folgenden Input-Informationen :

E1: Gewicht des Körpers = 7ø g;
E2: Scheinbares Gewicht des Körpers in der Prüfflüssigkeit (zu messen mit einer Federwaage) = 4ø g;
E3: Wichte der Prüfflüssigkeit = 1 (Wasser),

dann erzeugt das Programm die folgenden Ergebnisse :

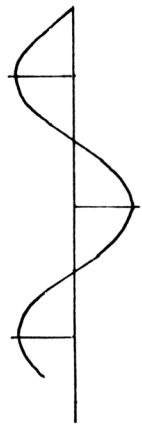
```
ZU PRUEFENDER KOERPER :

     VOLUMEN      =  3ø

     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.

(2.) SCHRITT : Problemanalyse

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

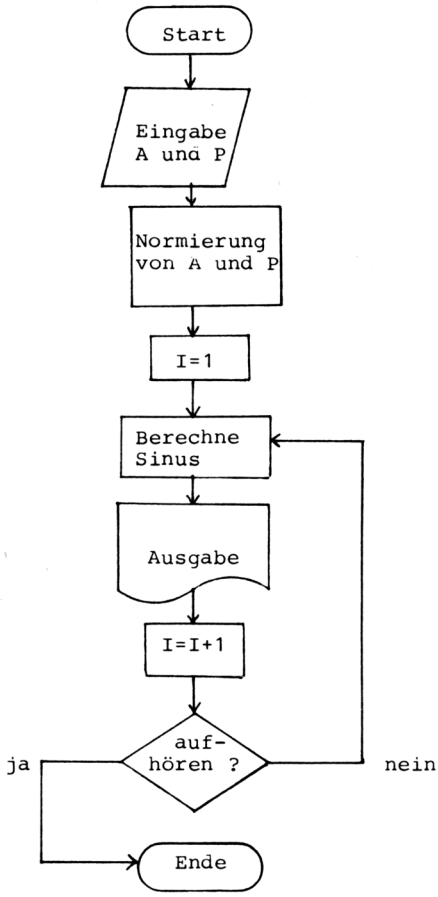
Das Programm kann dann veranlassen, daß - wie in obiger Skizze - die Sinusschwingung von oben nach unten über den Bildschirm "läuft" (was zu einem einfacheren Programm führt, als wenn diese Schwingung horizontal ausgegeben werden sollte).

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 2ø legen und die Amplitude, also den größtmöglichen Ausschlag z.B. auf den Maximalwert 19 ausrichten.

Wir wählen hier also - wie aus der gerade genannten Bildschirmaufteilung hervorgeht - nicht die hochauflösende Graphik (dazu siehe Kap. 9), sondern stellen die Schwingung in der sog. Blockgraphik (Normalgraphik) unter Nutzung eines geeigneten ASCII-Codezeichens dar.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM P3-PENDEL
20 CLS
30 PRINT"PROGRAMM ZUR SIMULATION EINER PENDEL-"
40 PRINT TAB(14)"BEWEGUNG.":PRINT:PRINT
50 PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984":PRINT
52 PRINT:PRINT"ZUM UNTERBRECHEN ESC-TASTE DRUECKEN
54 PRINT:PRINT"ZUM ENDGUELTIGEN ANHALTEN ZWEIMAL E
SC"
56 PRINT TAB(10)"TASTE DRUECKEN !"
60 INPUT "AMPLITUDE DER SCHWINGUNG : ";A
70 PRINT:INPUT"PERIODENLAENGE              : ";P
75 IF A>19 THEN A=19
80 CLS
90 I=1
100 J=(2*(22/7)/P)*I
110 Y=SIN(J)
120 S=20+Y*A
130 FOR K=1 TO S
140 PRINT CHR$(231);
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)*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)

6. SCHRITT : Programmbeschreibung

Satz 1ø-55 : Überschrift und Erläuterungen

Satz 6ø-7ø : Anforderung der Input-Informationen

Satz 75 : Beschränkung der Schwingungsamplitude auf den Maximalwert 19

Satz 8ø : Löschen des Bildschirms

Satz 9ø : Beginn Ausdruckschleife (I = Zeilenzählindex)

Satz 1øø : Normierung von I nach Maßgabe der Periodenlänge P zur Hilfsgröße J

Satz 11ø : Bestimmung des dazugehörigen Sinuswerts

Satz 12ø : Bestimmung der Bildschirmzeilen-Position des jeweiligen Sinuswertes unter Berücksichtigung der Tatsache, daß der Nullwert in der 2ø.Spalte liegen soll

Satz 13ø-15ø : Schleife zum Auffüllen der Zeile bis zum jeweiligen Sinuswert mit "Kügelchen"

Satz 16ø-17ø : Erzwingen eines Zeilenvorschubs und Ausgabe der nächsten Zeile durch Rücksprung zu Satz 1øø nach Erhöhung von I um 1

(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 gelben "Kügelchen" gefüllt werden.

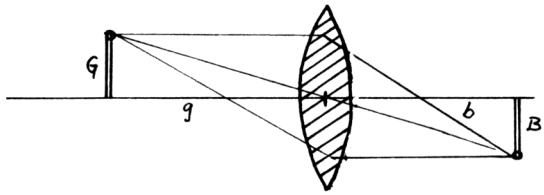
Der Leser kann sich leicht selbst ausdenken, wie das Programm verändert werden müßte, wenn man dieses "Auffüllen" unterdrücken wollte, also nur diejenigen Positionen z.B. durch "Kügelchen" optisch markiert, die tatsächlich auf der Sinuslinie liegen.

Das hier vorgestellte Programm ist ein Endlosprogramm, das nur durch Betätigung der [ESC] -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. Gegenstandsentfernung,
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 :

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

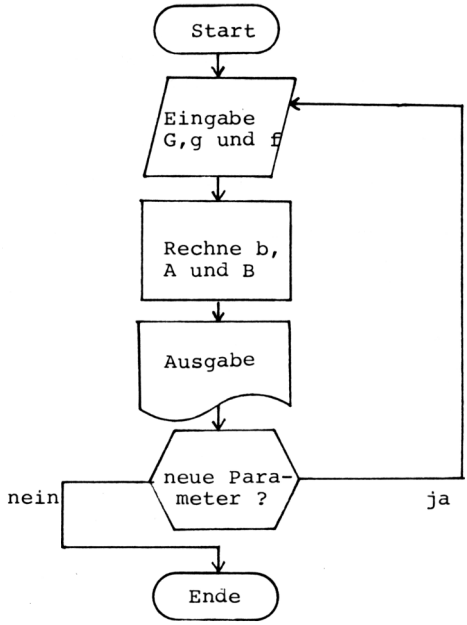
Weiterhin gilt, daß der Abbildungsmaßstab A sich ergibt zu :

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

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

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

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM P4-LINSE
20 CLS
30 PRINT"PROGRAMM ZUR BERECHNUNG VON BILDWEITE,"
40 PRINT"BILDGROESSE UND ABBILDUNGSMASSTAB BEI"
50 PRINT TAB(9)"EINER KONVEXEN LINSE.":PRINT
60 PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984":PRINT
70 PRINT:PRINT"EINZUGEBEN SIND DIE FOLGENDEN DATEN
 :":PRINT
80 PRINT TAB(5)"- GEGENSTANDSGROESSE : ";:INPUT G
90 PRINT TAB(5)"- GEGENSTANDSWEITE   : ";:INPUT GW
100 PRINT TAB(5)"- BRENNWEITE        : ";:INPUT F
110 CLS
120 BW=1/(1/F-1/GW)
130 A=BW/GW
140 B=(G*BW)/GW
145 BW=INT(BW*100+0.5)/100:A=INT(A*100+0.5)/100:B=
INT(B*100+0.5)/100
150 PRINT"AUSGANGSDATEN : ":PRINT
160 PRINT TAB(5)"BRENNWEITE          : ";F
170 PRINT:PRINT TAB(5)"GEGENSTANDSGROESSE   : ";G
180 PRINT:PRINT TAB(5)"GEGENSTANDSWEITE     : ";GW
190 PRINT:PRINT:PRINT"ERGEBNISSE : ":PRINT
200 PRINT:PRINT TAB(5)"ABBILDUNGSMASSTAB    : ";A
210 PRINT:PRINT TAB(5)"BILDGROESSE          : ";B
220 PRINT:PRINT TAB(5)"BILDWEITE            : ";BW
230 PRINT:PRINT:PRINT:INPUT"NOCH EINE RECHNUNG (J/
N) ";A$
240 IF A$="J" THEN CLS: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

6. SCHRITT : Programmbeschreibung

Satz 1ø-7ø : Überschrift und Erläuterungen

Satz 8ø-1øø : Anforderung der Inputinformationen

Satz 11ø-14ø : Löschen des Bildschirms und Berechnungen

Satz 145 : Runden auf 2 Dezimalstellen

Satz 15ø-22ø : Ausgabe der Ergebnisse

Satz 23ø-24ø : Abfrage, ob weitere Berechnung gewünscht (dann zurück nach Satz 7ø nach Löschen des Bildschirms)

Satz 25ø : Beendigung des Programms

7. SCHRITT : Ergebnisse

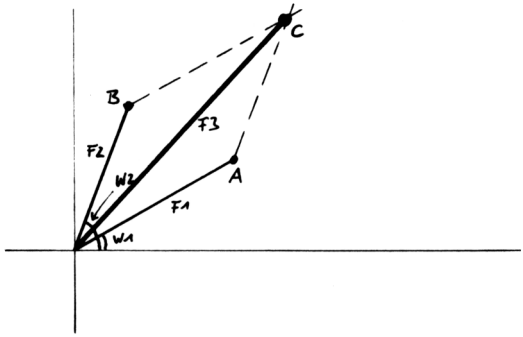
Geben wir z.B. ein : G = 2ø cm, GW = 3øø cm, F = 15 cm, so erhalten wir :

```
     BRENNWEITE              : 15
     GEGENSTANDSGROESSE      : 2ø
     GEGENSTANDSWEITE        : 3øø
ERGEBNISSE :
     ABBILDUNGSMASSTAB       : .ø5
     BILDGROESSE             : 1.ø5
     BILDWEITE               : 15.79
NOCH EINE RECHNUNG   (J/N)   ?
```

4.8 Kräfteparallelogramm

1. SCHRITT : Vorstellung des Problems

Wenn zwei Kräfte an einem gemeinsamen Punkt ansetzen, so ergibt sich eine gemeinsame resultierende Kraft als Diagonale des sog. Kräfteparallelogramms, wie die folgende Skizze verdeutlicht:



Im folgenden Programm soll für die beliebige Konstellation von je zwei Kräften das Kräfteparallelogramm graphisch ausgegeben und die resultierende Kraft F3 berechnet werden.

(2.) SCHRITT : Problemanalyse

Der Ansatzpunkt der beiden Kräfte soll in die Mitte des Bildschirms gelegt werden (32ø,2øø).

Einzugeben sind die Kräfte F1 und F2, die zunächst so umzurechnen sind, daß F1 + F2 nicht größer als 32ø wird, damit auch bei ungünstigster Konstellation der Platz des Bildschirms ausreicht.

Einzugeben sind weiterhin die Winkel W1 und W2, in denen die Kräfte F1 und F2 am Ansatzpunkt angreifen.

Mit Hilfe des Satzes des Pythagoras und der Benutzung der Winkelbeziehungen im rechtwinkligen Dreieck können dann Spalten und Zeilen von A und B und daraus dann auch die Koordinaten von C berechnet werden (siehe Skizze).

Ist die Lage von C bekannt, kann wiederum aus F1 und F2 die interessierende Kraft F3 berechnet werden

(3.) SCHRITT : Flußdiagramm

Auf ein Flußdiagramm kann hier verzichtet werden.

4. SCHRITT : Programm

```
10 REM P5-PARALLELOGRAMM
20 CLS
30 PRINT"PROGRAMM ZUR VERANSCHAULICHUNG DES SOG."
40 PRINT" KRAEFTE-PARALLELOGRAMMS BEI ZWEI AN-"
50 PRINT TAB(9)"GREIFENDEN KRAEFTEN.":PRINT:PRINT
60 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984":
PRINT
70 SU=320:ZU=200
80 PRINT:PRINT:PRINT:PRINT
90 INPUT "ERSTE KRAFT     F1           : ";F1
100 INPUT "ANSATZWINKEL    F1 (0-360)   : ";W1
105 PRINT:PRINT
110 INPUT "ZWEITE KRAFT    F2           : ";F2
120 INPUT "ANSATZWINKEL    F2 (0-360)   : ";W2
130 W1=W1*((22/7)/180):W2=W2*((22/7)/180)
135 H=F1:G=F2
140 F1=(F1/(F1+F2))*300:F2=(F2/(H+F2))*300
150 Z1=F1*SIN(W1)+200
160 S1=((Z1-200)/TAN(W1))+320
165 CLS
170 PLOT 320,200
180 DRAW S1,Z1
190 Z2=F2*SIN(W2)+200
200 S2=((Z2-200)/TAN(W2))+320
210 PLOT 320,200
220 DRAW S2,Z2
230 S3=S2+(S1-SU)
240 Z3=Z2+(Z1-ZU)
250 PLOT SU,ZU
260 DRAW S3,Z3
270 F3=SQR((Z3-ZU)^2 + (S3-SU)^2)
275 F3=F3*(G+H)/300
280 LOCATE 5,16
290 PRINT"KRAFT 1  =  ";H
300 PRINT TAB(5)"KRAFT 2  =  ";G
310 PRINT:PRINT"RESULTIERENDE KRAFT  = ";F3
320 PRINT:PRINT:PRINT:INPUT "NOCHMAL (J/N) ";A$
330 IF A$="J" THEN CLS: GOTO 90
340 PRINT:PRINT:PRINT"ENDE":END
```

5. SCHRITT : Variablenliste

AØ = Stringvariable (Ja/Nein)
F1 = erste Kraft
F2 = zweite Kraft
F3 = dritte Kraft
G = Hilfsgröße (=F2)
H = Hilfsgröße (=F1)
SU = Spalte Ursprung
S1 = Spalte Punkt A
S2 = Spalte Punkt B
S3 = Spalte Punkt C
W1 = erster Ansatzwinkel
W2 = zweiter Ansatzwinkel
ZU = Zeile Ursprung
Z1 = Zeile Punkt A
Z2 = Zeile Punkt B
Z3 = Zeile Punkt C

6. SCHRITT : Programmbeschreibung

Satz 1Ø- 6Ø : Überschrift

Satz 7Ø : Ursprungskoordinaten

Satz 8Ø-12Ø : Eingabe der Ausgangsinformationen

Satz 13Ø : Umrechnen der Winkel in Einheiten des Kreisparameters

Satz 135 : Speichern der eingegebenen Kraftangaben

Satz 14ø : Normierung der beiden Kräfte zur Bildschirmausnutzung

Satz 15ø-16ø : Bestimmung der Koordinaten von A

Satz 165-18ø : Zeichnen von F1

Satz 19ø-2øø : Koordinaten von B

Satz 21ø-22ø : Zeichnen von F2

Satz 23ø-24ø : Koordinaten von C

Satz 25ø-26ø : Zeichnen von F3

Satz 27ø : Berechnen von F3 (Pythagoras!)

Satz 275 : Rückgängigmachen der Normierung von Satz 14ø für F3

Satz 28ø-31ø : Ausgabe der Ergebnisse

Satz 32ø-34ø : Beendigung des Programms, es sei denn, es wird eine Alternativrechnung gewünscht (dann zurück zu Satz 9ø).

(7.) SCHRITT : Ergebnisse

Auf die Darstellung von Ergebnissen kann hier - angesichts der Skizze aus dem 1. Schritt - verzichtet werden.

4.9 Das Ohm'sche Gesetz

(1.) SCHRITT : Vorstellung des Porblems

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.

Diese Aufgabe gibt uns die Gelegenheit, die PRINT USING-Anweisung einzuführen:

Der Satz

```
X = 12.323:PRINT USING "  .   ";X
```

führt zum Ergebnis 12.32.

Mit der "Maske" " " (oder anderen) kann also angegeben werden, mit wievielen Stellen ein Rechenergebnis erscheinen soll. Es wird dabei korrekt gerundet.

(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.

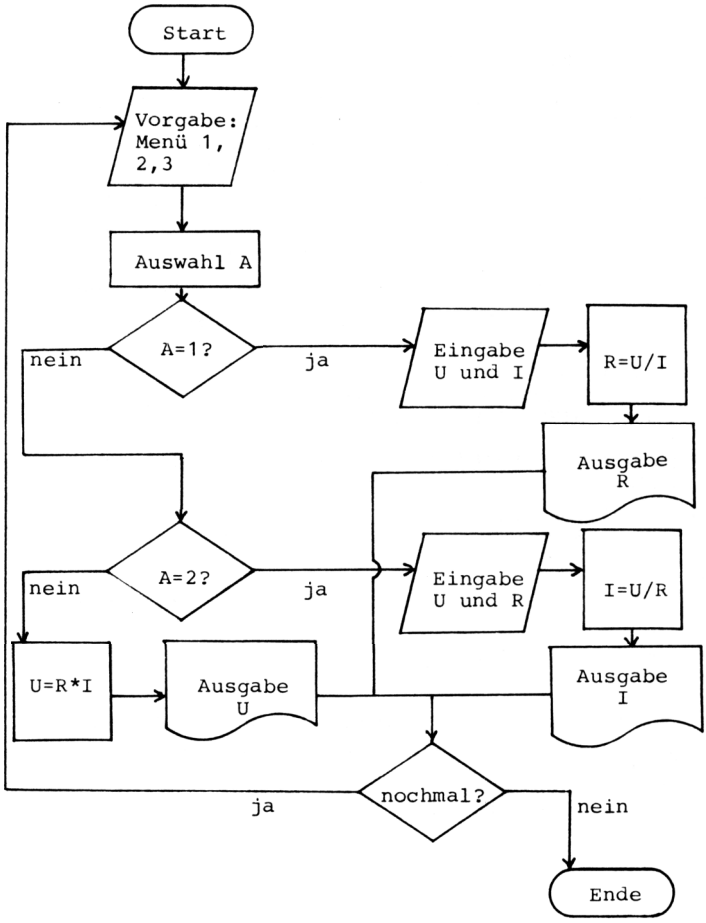
Dies gibt uns Gelegenheit, die sog. Menütechnik vorzuführen :

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



(4.) SCHRITT : Programm

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

5. SCHRITT : Variablenliste

A = Auswahlvariable (Menü)
A$ = Stringvariable (Ja,Nein)

I = Stromstärke
R = Widerstand
U = Spannung

6. SCHRITT : Programmbeschreibung

Satz 1ø-6ø : Überschrift und Erläuterungen

Satz 7ø-1øø : Vorgabe des Menüs

Satz 11ø : Menüauswahl

Satz 12ø-13ø : Wenn A größer als 1 ist (A = Menü-variable) geht es nach Löschen des Bildschirms weiter bei Satz 19ø

Satz 14ø-15ø : Eingaben gemäß Auswahl A = 1

Satz 16ø-17ø : Berechnungen und Ergebnisausgabe

Satz 18ø : Sprung zum Satz 3øø

Satz 19ø-24ø : Entsprechende Berechnungen wie oben; nun aber für A = 2

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

Satz 3øø-31ø : Abfrage, ob noch eine Berechnung gewünscht wird

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

Satz 32ø : Beendigung des Programms

(7.) SCHRITT : Ergebnisse

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

WIDERSTAND : 2ø

NOCH EINE RECHNUNG (J/N) ?

Kapitel 5 : Sprachen

5.1 Vorbemerkung und BASIC-Ergänzungen

Auch beim Erlernen oder beim Üben von Fremdsprachen können Rechner wirkunsvolle 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 20:

```
nn  Name$=INKEY$
```

Dieses Statement wartet, ähnlich wie das schon oft benutzte INPUT-Statement, eine Informationseingabe ab, die dann der Stringvariablen Name$ zugeordnet wird. Diese Eingabe braucht nicht durch die ENTER-Taste abgeschickt zu werden. In den späteren Programmen wird die Funktionsweise deutlich werden.

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 21:

```
nn GOSUB mm
```

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

Ist dann das Unterprogramm abgearbeitet, muß ein Rücksprung in das Hauptprogramm erfolgen. Zu diesem Zweck wird das Unterprogramm mit dem folgenden Statement beschlossen :

Statement 22:

```
nn RETURN
```

Dieses Statement bewirkt, daß ein Rücksprung an diejenige Stelle des Hauptprogramms erfolgt, die direkt hinter der entsprechenden GOSUB-Anweisung folgt.

Auch die Wirkungsweise dieser beiden Statements wird in den folgenden Programmen deutlich werden, so daß hier auf Demonstrationsbeispiele verzichtet werden kann.

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 2ø 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.

(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 2ø Verben aufgenommen),

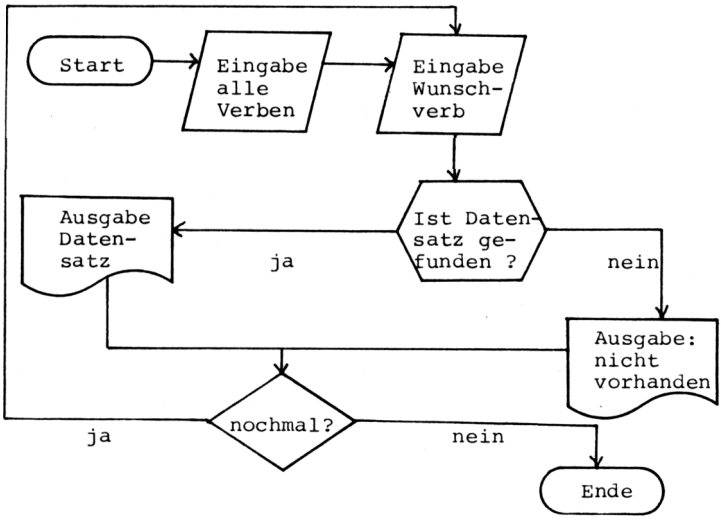
- oder, weil es sich bei dem eingegebenen Verb nicht um ein unregelmäßiges Verb handelt.

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



4. SCHRITT : Programm

```
10 REM S1-UNREGELM.VERBEN,ENGLISCH
20 CLS
30 PRINT"PROGRAMM ZUR AUSGABE DER STAMMFORMEN"
40 PRINT"VON EINGEGEBENEN ENGLISCHEN UNREGEL-"
50 PRINT TAB(11)"MAESSIGEN VERBEN."
60 PRINT:PRINT:PRINT TAB(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 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !"
140 A$=INKEY$
150 IF A$="" THEN 140
155 CLS:PRINT TAB(12)"MOMENT BITTE":PRINT:PRINT
160 N=20
170 DIM W$(N,4)
180 FOR I=1 TO N:FOR J=1 TO 4:READ W$(I,J):NEXT J:
NEXT I
190 CLS
200 PRINT"BITTE DAS DEUTSCHE VERB, DESSEN ENGL-"
210 PRINT"SCHE STAMMFORMEN GEWUENSCHT WERDEN,"
220 INPUT "EINGEBEN : ";V$
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 V
ORHANDEN,"
270 PRINT"BZW. NICHT UNREGELMAESSIG.":GOTO 310
275 PRINT:PRINT:PRINT
280 PRINT"VERB : ";W$(I,1):PRINT:PRINT:PRINT
290 PRINT"ENGLISCHE STAMMFORMEN : ":PRINT
300 FOR J=2 TO 4:PRINT W$(I,J);"  ";:NEXT J
310 PRINT:PRINT:PRINT:INPUT"NOCHMAL (J/N) ";A$
320 IF A$="J" THEN CLS:GOTO 200
330 PRINT: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,DO,DID,DONE
512 DATA ESSEN,EAT,ATE,EATEN
514 DATA KAUFEN,BUY,BOUGHT,BOUGHT
516 DATA FUEHLEN,FEEL,FELT,FELT
```

```
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 für Eingaben

I = Laufvariable
J = Laufvariable
N = Anzahl der Datensätze

V$ = Gewünschtes Verb
W$ = Stammformen der Verben und deutsche Übersetzung (Ausgangsdatenbestand)

(6.) SCHRITT : Programmbeschreibung

Satz 1ø-135 : Überschrift, Erläuterungen und Hinweise, wie der Datenbestand aktualisiert werden kann

Satz 14ø-15ø : Programm in "Warteposition". Die Programmausführung geht erst weiter, wenn der Benutzer irgendeine Taste drückt.

Satz 155 : Löschen des Bildschirms und Hinweis

Satz 16ø-18ø : Dimensionierung und Einlesen der Daten

Satz 19ø-22ø : Löschen des Bildschirms und Anforderung des gewünschten Verbs

Satz 23ø-25ø : Suche nach dem richtigen Datenbestand

Wird er gefunden, dann weiter bei 275
wird er nicht gefunden, weiter bei 26ø

Satz 26ø-27ø : Meldung, daß Verb im Datenbestand nicht vorhanden; dann weiter bei Satz 31ø

Satz 275-3øø : Ausgabe der gesuchten Informationen

Satz 31ø-32ø : Abfrage, ob weitere Ausgabe gewünscht

wenn ja, Löschen des Bildschirms und zurück zu Satz 2øø

Satz 33ø : wenn nein, Ende des Programms

Satz 5øø-538 : Daten

(7.) SCHRITT : Ergebnisse

Gibt man beispielweise auf die Anforderung des Rechners aus Satz 2øø ein :

GO

so antwortet der Rechner :

VERB : GEHEN

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 5øø ff. sind weitere DATA-Statements anzuhängen und Satz 16ø muß ausgetauscht werden mit der Angabe der aktuellen Zahl eingegebener Verben.

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 5.2 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.

2. SCHRITT : Problemanalyse

Bei diesem Problem können wir uns kurz fassen :

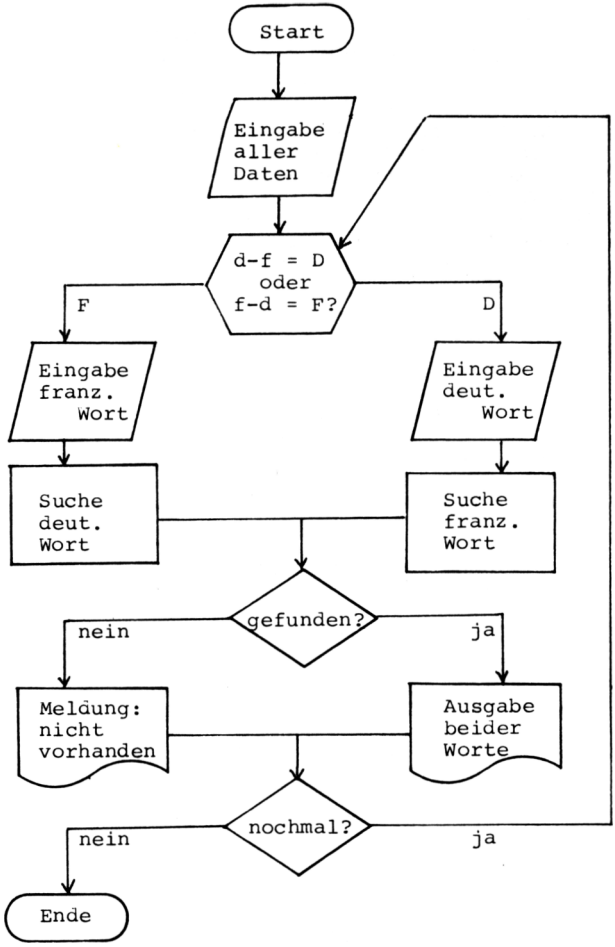
Gibt der Benutzer ein deutsches Wort ein, so muß der Rechner durch Vergleich mit allem ihm zur Verfügung stehenden deutschen (und damit paarweise verbundenen französischen Worten) das entsprechende deutsch-französische Wortpaar aus seinem Datenbestand heraussuchen und ausgeben. Entsprechend ist auch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



4. SCHRITT : Programm

```
10 REM S2-VERBEN DEUTSCH-FRANZOESICH
20 CLS
30 PRINT"PROGRAMM ZUM NACHSCHLAGEN VON VOKABELN:"
40 PRINT:PRINT"DEUTSCH/FRANZOESICH ODER UMGEKEHRT.
50 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
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 GEAENDERT WERDEN."
130 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !"
140 A$=INKEY$:IF A$="" THEN 140
150 CLS:PRINT"MOMENT BITTE"
160 N=10
170 DIM D$(N),F$(N)
180 FOR I=1 TO N:READ D$(I),F$(I):NEXT I
190 CLS
200 PRINT"WELCHE UEBERSETZUNG WIRD GEWUENSCHT ?"
210 PRINT:PRINT:PRINT"- DEUTSCH-FRANZOESICH  (1)"
220 PRINT:PRINT"ODER"
230 PRINT:PRINT  "- FRANZOESICH-DEUTSCH  (2)"
240 PRINT:PRINT:INPUT"BITTE 1 ODER 2 EINGEBEN ";Z
245 CLS
250 IF Z=2 THEN 350
260 PRINT:PRINT:INPUT"DEUTSCHES WORT BITTE : ";W$
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 CLS:PRINT"DEUTSCH          : ";D$(I)
330 PRINT:PRINT  "FRANZOESISCH      : ";F$(I)
340 GOTO 450
350 PRINT:PRINT:INPUT"FRANZ. WORT BITTE : ";W$
360 FOR I=1 TO N:IF W$=F$(I) THEN 400
370 NEXT I
380 PRINT:PRINT"GESUCHTES WORT IM DATENBESTAND NICHT
390 PRINT"VORHANDEN.":GOTO 450
400 CLS:PRINT"FRANZOESICH   : ";F$(I)
410 PRINT:PRINT  "DEUTSCH       : ";D$(I)
450 PRINT:PRINT:PRINT:INPUT"NOCHMAL ? (J/N) ";A$
460 IF A$="J" THEN CLS:GOTO 200
470 PRINT:PRINT"ENDE DER AUSGABE":END
```

```
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
```

5. SCHRITT : Variablenliste

A$ = Stringvariable für Eingaben

D$ = deutsche Vokabeln
F$ = französische Vokabeln

I = Laufindex
N = Anzahl der Datensätze

W$ = gewünschtes Wort

Z = Kennziffer bei der Menü-Auswahl

6. SCHRITT : Programmbeschreibung

Satz 1ø-12ø : Überschrift, Erläuterungen etc.

Satz 13ø-15ø : Warten, bis Tastendruck erfolgt und Löschen des Bildschirms

Satz 16ø-18ø : Angabe der Anzahl der Datensätze, Dimensionierung und Eingabe der Ausgangsdaten

Satz 19ø-24ø : Löschen des Bildschirms und Menüvorgabe

Satz 245-25ø : Wird 2 eingegeben (französisch-deutsch), dann weiter bei Satz 35ø

Satz 26ø : Eingabe eines deutschen Wortes

Satz 28ø-29ø : Suche nach dem entsprechenden Datensatz

Satz 3øø-31ø : Ist das gesuchte Wort im Datenbestand (noch) nicht vorhanden, weiter bei 45ø

Satz 32ø-33ø : Ausgabe der Ergebnisse

Satz 34ø : Sprung nach Satz 45ø

Satz 35ø-41ø : Entsprechende Vorgehensweise für die Eingabe eines französischen Wortes

Satz 45ø-47ø : Beendigung des Programms mit Wiederholungsmöglichkeit

Satz 5øø-518 : Daten

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

FRANZÖSISCH : AVOIR

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.

(2.) SCHRITT : Problemanalyse

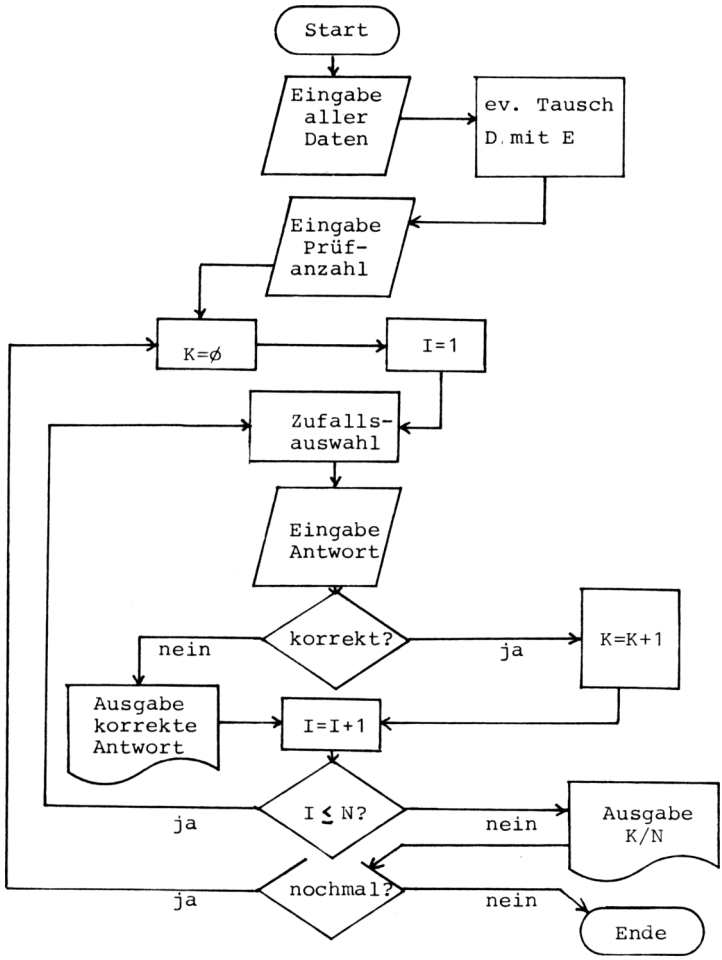
Ausgehend von einem vorgegebenen Datenbestand, muß das Programm eine Zufallsauswahl treffen.

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



(4.) SCHRITT : Programm

```
10 REM S3-VOKABELTEST
20 CLS
30 PRINT"PROGRAMM ZUM ABFRAGEN VON VOKABELN."
40 PRINT:PRINT"HIER : DEUTSCH/ENGLISCH ODER UMGEKE
HRT"
50 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
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 1000 FF.WEITERE DATA-STA-"
110 PRINT"TEMENTS ANGEFUEGT UND DER WERT FUER N"
120 PRINT"IN STATEMENT 160 GEAENDERT WERDEN."
130 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !"
140 A$=INKEY$:IF A$="" THEN 140
150 CLS:PRINT"MOMENT BITTE"
160 N=10
170 DIM D$(N,2)
180 FOR I=1 TO N:READ D$(I,1),D$(I,2):NEXT I
185 ZZ=1
190 CLS
200 PRINT"DEUTSCH/ENGLISCH      (1)"
210 PRINT:PRINT:PRINT"ODER"
220 PRINT:PRINT:PRINT"ENGLISCH/DEUTSCH      (2)"
230 PRINT:PRINT:PRINT"BITTE 1 ODER 2 EINGEBEN ":PR
INT
240 INPUT"                          ";Z
245 IF Z<>ZZ THEN GOSUB 3000:REM TAUSCH
250 PRINT:PRINT:PRINT"WIEVIELE VOKABELN SOLLEN GEP
RUEFT"
260 INPUT "DEN ";A
270 GOSUB 2000
400 PRINT:INPUT "NOCHMAL ? (J/N) ";A$
410 IF A$="J" THEN CLS:ZZ=Z:GOTO 200
420 PRINT:PRINT"ENDE DER AUSGABE":END
450 PRINT:PRINT:PRINT:INPUT"NOCHMAL ? (J/N) ";A$
460 IF A$="J" THEN CLS:GOTO 200
470 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,TAST
E,KEY
```

```
2000 CLS
2005 K=0
2010 I=1
2020 R=INT(RND(1)*N+1)
2030 PRINT D$(R,1);:INPUT"    ANTWORT    : ";W$
2040 IF W$=D$(R,2) THEN K=K+1:PRINT:PRINT:PRINT TAB(10)"SEHR GUT !":GOTO 2070
2050 PRINT:PRINT:PRINT"DAS WAR LEIDER FALSCH."
2060 PRINT:PRINT"RICHTIG MUSS ES HEISSEN : ";D$(R,2)
2070 PRINT:PRINT:PRINT:I=I+1
2080 IF I<=A THEN 2020
2090 KA=(K/A)*100
2100 PRINT:PRINT"ANTEIL KORREKTER ANTWORTEN : ";KA;"%"
2110 RETURN
3000 REM UP TAUSCH
3010 FOR I=1 TO N
3020 H$=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

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

6. SCHRITT : Programmbeschreibung

Satz 1ø-12ø : Überschrift, Erläuterungen und Hinweise auf die eventuelle Verlängerung des Programms

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

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

Satz 185 : Belegung der Hilfsvariablen ZZ mit 1

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

Satz 245 : Abfrage, ob Menü 1 gewählt wurde

wenn nein, Sprung ins UP 3øøø
wenn ja, weiter bei 25ø

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

Satz 27ø : Sprung ins UP 2øøø

Satz 4øø-42ø : Abfrage, ob noch ein Test (zurück zu Satz 2øø) und Beendigung des Programms

Satz 1øøø-1ø2ø: Daten

Satz 2ØØØ-211Ø : UP Zufallsauswahl :

2ØØ5-2Ø1Ø	: Zähler K und I auf Anfangspositionen
2Ø2Ø	: Zufallszahl
2Ø3Ø	: Ausgabe der Vokabel und Anforderung der Antwort
2Ø4Ø	: Bei korrekter Antwort erfolgt Sprung nach 2Ø7Ø
2Ø5Ø-2Ø6Ø	: Fehlermeldung und Ausgabe der korrekten Antwort
2Ø7Ø-2Ø8Ø	: Nächste Vokabel
2Ø9Ø-21ØØ	: Ausgabe des Erfolgsanteils
211Ø	: Rücksprung

Satz 3ØØØ-3Ø4Ø : 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 2Ø4Ø bis Satz 21ØØ).

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.

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.

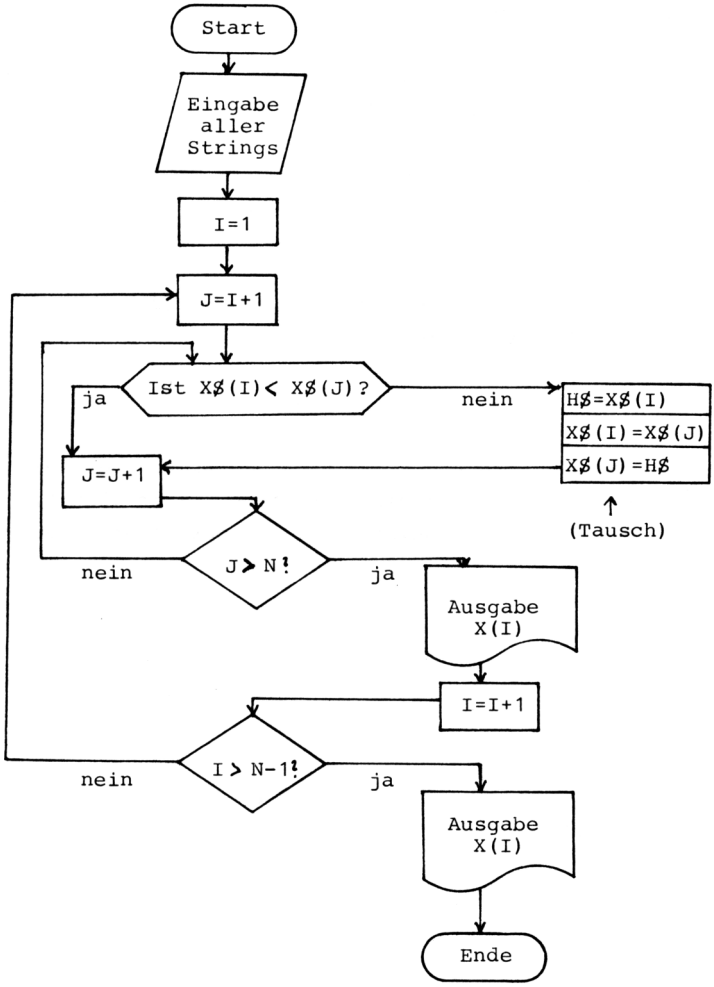
Es werden insgesamt so viele Runden durchlaufen, wie überhaupt Vergleiche möglich sind. Haben wir beispielsweise fünf Strings zu sortieren, so müssen vier derartige Runden durchlaufen werden : In der ersten Runde gibt es vier Vergleiche, in der zweiten Runde drei, in der dritten Runde zwei und in der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

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



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 Schreibweis e oder nach der englischen Schreibweise sortieren möchte :

```
10 REM S4-SORT
20 CLS
30 PRINT TAB(2)"PROGRAMM ZUM SORTIEREN VON VOKABEL
N."
40 PRINT:PRINT:PRINT TAB(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 GEAENDERT WERDEN."
110 LOCATE 5,23
120 PRINT"BITTE EINE TASTE DRUECKEN !"
130 A$=INKEY$:IF A$="" THEN 130
140 CLS
150 N=10
160 DIM E$(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$="D" THEN 240
220 CLS:PRINT"ENGLISCH";TAB(20)"DEUTSCH":PRINT:PRI
NT
230 GOSUB 1000:REM UP SORT
235 GOTO 270
240 FOR I=1 TO N:H$=E$(I):E$(I)=D$(I):D$(I)=H$:NEX
T I
250 CLS:PRINT"DEUTSCH";TAB(20)"ENGLISCH":PRINT:PRI
NT
260 GOSUB 1000:REM UP SORT
270 PRINT:PRINT"ENDE DER AUSGABE":END
```

```
500 DATA PRINT,DRUCKEN,WRITE,SCHREIBEN,END,ENDE,GO
,GEHEN
510 DATA RUN,LAUFEN,DATA,DATEN,SUBROUTINE,UNTERPRO
GRAMM
520 DATA INPUT,EINGABE,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 E$(I);TAB(20)D$(I)
1090 NEXT I
1100 PRINT E$(I);TAB(20)D$(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

6. SCHRITT : Programmbeschreibung

Satz 1ø-11ø : Überschrift und Erläuterungen

Satz 12ø-14ø : Warten und Löschen des Bildschirms

Satz 15ø-17ø : Angabe der Zahl der Vokabeln, Dimensionierung und Einlesen der Vokabeln

Satz 18ø-21ø : Abfrage, ob nach deutschen oder nach englischen Worten sortiert werden soll

falls nach deutschen Worten, weiter bei 27ø; andernfalls weiter bei 22ø

Satz 22ø : Ausgabe einer Tabellenüberschrift

Satz 23ø : Sprung ins UP 1øøø zum Sortieren

Satz 235 : Sprung zum Satz 27ø

Satz 24ø : Austausch der deutschen und der englischen Vokabeln

Satz 25ø : Ausgabe einer Tabellenüberschrift

Satz 26ø : Sprung ins UP 1øøø zum Sortieren

Satz 27ø : Beendigung des Programms

Satz 5øø-52ø : Daten

Satz 1ØØØ-111Ø : UP Sortieren :

Unterprogramm zum Sortieren von Strings :

1Ø1Ø	: Rundenzähler
1Ø2Ø	: Vergleichszähler
1Ø3Ø	: Vergleich fällt er befriedigend aus, weiter bei 1Ø7Ø, sonst weiter bei 1Ø4Ø
1Ø4Ø-1Ø6Ø	: Tausch
1Ø7Ø	: Nächster Vergleich
1Ø8Ø	: Nachdem kein Vergleich mehr möglich ist, Ausgabe der beiden nun an erster Stelle stehenden Strings
1Ø9Ø	: Nächste Runde
11ØØ	: Ausgabe des letzten Stringpaares
111Ø	: 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
...	...

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 Xø 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 X1 :

$$X1 = X\emptyset + X\emptyset * R/1\emptyset\emptyset = X\emptyset * (1 + R/1\emptyset\emptyset)$$

Nach zwei Jahren ergibt sich X2 :

$$X2 = X1 * (1 + R/1\emptyset\emptyset) = X\emptyset * (1+R/1\emptyset\emptyset) * (1+R/1\emptyset\emptyset)$$

Daraus folgt :

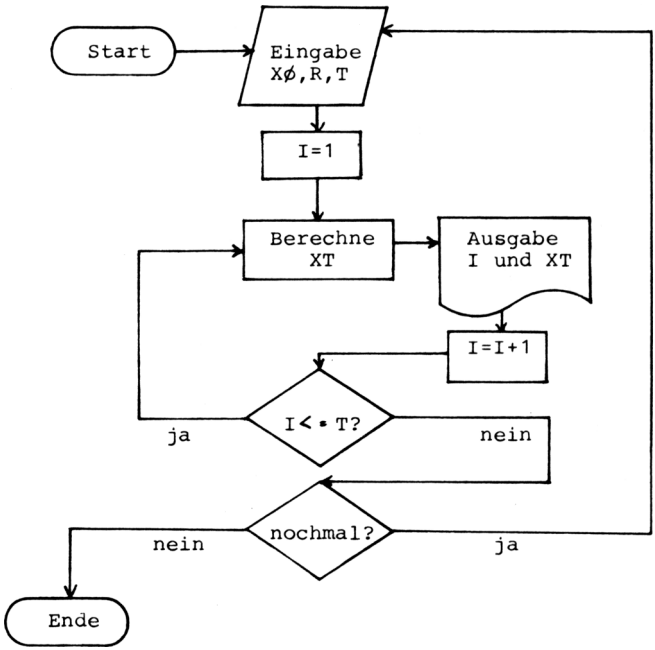
$$X2 = X\emptyset * (1 + R/1\emptyset\emptyset)^2$$

Generell gilt deshalb nach T Jahren :

$$XT = X\emptyset * (1 + R/1\emptyset\emptyset)^T$$

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

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM B1-WACHSTUM
20 CLS
30 PRINT"PROGRAMM ZUR TABELLARISCHEN DARSTELLUNG"
40 PRINT TAB(8)"UNGEBREMSTEN WACHSTUMS."
50 PRINT:PRINT:PRINT TAB(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 CLS:PRINT"NACH ... JAHREN    BESTAND":PRINT:PR
INT
110 I=1
120 XT=X0*(1+R/100)^I
130 PRINT TAB(5)I;TAB(19) USING "######.##";XT
140 I=I+1
145 IF I/10 = INT(I/10) THEN PRINT:PRINT"BITTE CON
T EINGEBEN":STOP
150 IF I<=T THEN 120
160 PRINT:PRINT:INPUT"NOCHMAL ? (J/N) ";A$
170 IF A$="J" THEN CLS: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

(6.) SCHRITT : Programmbeschreibung

Satz 1ø-6ø : Überschrift

Satz 7ø-9ø : Vorgabe der Input-Informationen, die das Programm benötigt

Satz 1øø : Ausgabe einer Tabellenüberschrift

Satz 11ø : Vorgabe der ersten Periode

Satz 12ø : Berechnung des jeweiligen Bestandes

Satz 13ø : Ausgabe

Satz 14ø : Übergang zur nächsten Periode

Satz 145 : Nach jeder zehnten Periode wird das Programm unterbrochen (also dann, wenn I durch 1ø ohne Rest teilbar ist); Fortsetzung ist nur mit dem Kommando CONT möglich, deshalb ein entsprechender Hinweis

Satz 15ø : Solange I kleiner gleich T ist, ist das Programm fortzusetzen (Satz 12ø)

Satz 16ø-18ø : Beendigung des Programms, es sei denn, es wird ein zusätzlicher Programmlauf gewünscht (dann nach Löschen des Bildschirms zurück zu Satz 7ø)

7. SCHRITT : Ergebnisse

Geben wir z.B. als Anfangsbestand den Wert 5øø 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              52ø
       2              54ø.8
       3              562.43
       4              584.93
       5              6ø8.33
       6              632.66
       7              657.97
       8              684.28

NOCHMAL    (J/N)   ?
```

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.

(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 Zuwachsrate mit fortschreitender Zeit absinkt. Beispielsweise können wir die Zuwachsrate bei Verdopplung der Zeit halbieren, bei Vervierfachung der Zeit auf ein Viertel zurückgehen lassen usw.

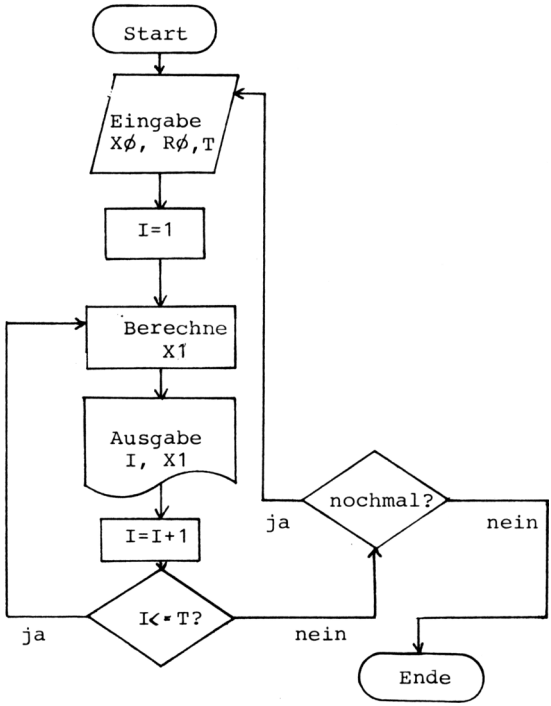
Natürlich lassen sich auch andere mathematische Modelle vorstellen, um diesen Bremsvorgang zu simulieren - darauf kommt es aber ja in diesem Zusammenhang nicht entscheidend an.

Den so beschriebenen Rechenalgorithmus können wir mathematisch folgendermaßen fassen :

$$X1 = X\emptyset * (1 + R/I)$$

Das heißt, der Bestand der folgenden Periode X1 berechnet sich aus dem der vorhergehenden Periode X$\emptyset$ 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



4. SCHRITT : Programm

```
10 REM B2=GEBREMSTES WACHSTUM
20 CLS
30 PRINT"PROGRAMM ZUR TABELLARISCHEN DARSTELLUNG"
40 PRINT TAB(2)"EINES GEBREMSTEN WACHSTUMSPROZESSES."
50 PRINT:PRINT:PRINT TAB(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 CLS
100 PRINT"NACH ... JAHREN   BESTAND   ZUWACHS (%)":PRINT:PRINT
110 I=1
120 X1=X0+X0*(R/I)/100
140 Z=(X1-X0)/X0:Z=Z*100
150 PRINT TAB(5)I;TAB(19) USING "######.##";X1;:PRINT TAB(28) USING "####.##";Z
160 X0=X1:I=I+1:IF I<=T THEN 120
180 PRINT:INPUT"NOCHMAL ? (J/N) ";A$
190 IF A$="J" THEN CLS: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

(6.) SCHRITT : Programmbeschreibung

Satz 1ø-6ø : Überschrift und Erläuterungen

Satz 7ø-9ø : Anforderung der Ausgangsdaten und Löschen des Bildschirms

Satz 1øø : Ausgabe einer Tabellenüberschrift

Satz 11ø : Vorgabe des Periodenzählers

Satz 12ø-15ø : Berechnung des Periodenbestands, des Zuwachses und Ausgabe der entsprechenden Werte nach Rundung auf zwei Dezimalstellen

Satz 16ø : Belegung von Xø mit X1 und Fortführung der Berechnungen, solange I nach Erhöhung um 1 kleiner als der vorgegebene Zeitwert T bleibt

Satz 18ø-2øø : Abfrage, ob noch eine Berechnung gewünscht (wenn ja, zurück nach 7ø) und Beendigung des Programms

7. SCHRITT : Ergebnisse

Geben wir zum Beispiel als Anfangsbestand den Wert 5øø 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	52ø	4
2	53ø.4	2
3	537.47	1.33
4	542.84	1
5	547.18	.8
6	55ø.83	.67
7	553.98	.57
8	556.75	.5

NOCHMAL (J/N) ?

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 kennengerlernt 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 ø jetzt nicht mehr als

$$X1 = Xø + Xø * R/1øø$$

sondern als :

$$X1 = Xø + Xø * (R/1øø - P*Bø)$$

Dabei ist Bø die Umweltbelastung zum Zeitpunkt ø, die über einen geeigneten Parameter P (z.B. in der Größenordnung P = ø.ø1) die Populationszuwächse bremst.

Natürlich brauchen wir jetzt aber auch eine funktionale Beziehung, welche die Größe B1 (Umweltbelastung in der Periode 1) in Abhängigkeit von Bø und Xø erklären kann.

Als Beispiel einer solchen Beziehung mag gelten :

$$B1 = Bø + A * Xø$$

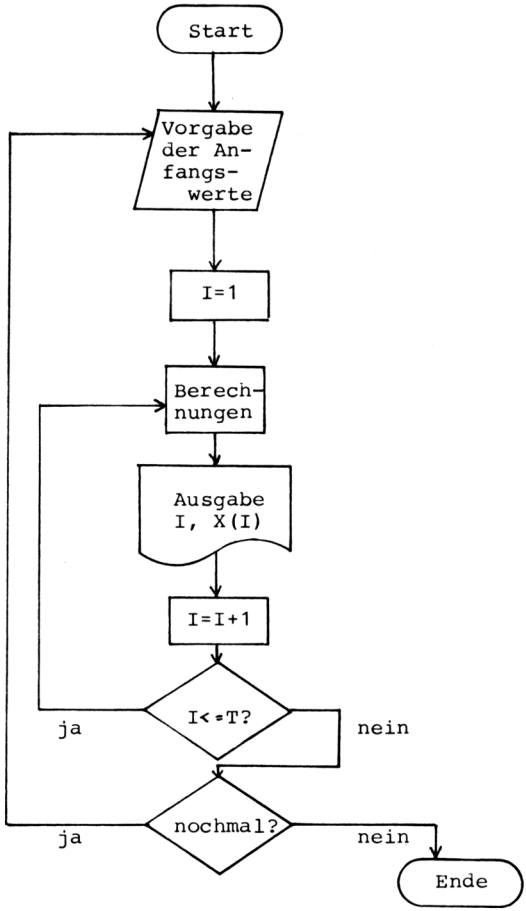
Von einer Belastungsgröße von z.B. Bø = ø ausgehend, wächst die Belastung in Abhängigkeit von der erreichten Populationsgröße nach Maßgabe eines geeignet festzusetzenden Parameters A (z.B. A = ø.ø15).

Somit gilt also der folgende Algorithmus :

(1) $B1 = Bø + A*Xø$

(2) $X1 = Xø + Xø * (R/1øø - P*B1)$

3. SCHRITT : Flußdiagramm



(4.) SCHRITT : Programm

```
10 REM B3-UMWELT
20 CLS
30 PRINT"PROGRAMM ZUR TABELLARISCHEN DARSTELLUNG"
40 PRINT TAB(2)"EINES WACHSTUMSPROZESSES, DER DURC
H"
50 PRINT TAB(2)"SEINE EIGENE DYNAMIK GEBREMST WIRD
."
60 PRINT:PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS,
1984"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"DIESES PROGRAMM BENOETIGT DIE FOLGENDEN"
90 PRINT"AUSGANGSINFORMATIONEN : ":PRINT
100 PRINT:INPUT"AUSGANGSBESTAND            : ";X0
105 PRINT:INPUT"WIEVIELE JAHRE             : ";T
110 PRINT:INPUT"JAEHRL.WACHSTUMSRATE (%)   : ";R
120 PRINT:INPUT"FAKTOR UMWELTBELASTUNG     : ";P
130 CLS
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
190 PRINT TAB(2)I;TAB(7) USING "######.##";X1;:PRIN
T TAB(17) USING "####.##";Z;
195 PRINT TAB(29) USING "######.##";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 CLS:GOTO 80
230 PRINT:PRINT"ENDE DER AUSGABE":END
```

5. SCHRITT : Variablenliste

A = Faktor populationsbedingte Umweltbelastung
A$ = Stringvariable (Ja, Nein)
B0 = Umweltbelastung Anfang
B1 = Umweltbelastung nächste Periode
I = Laufindex
P = Faktor Umweltbelastung
R = Jährliche Wachstumsrate
T = Zeit
X0 = Population Anfang
X1 = Population nächste Periode
Z = Populationszuwachs

6. SCHRITT : Programmbeschreibung

Satz 10-70 : Überschrift und Erläuterungen

Satz 80-120 : Anforderung der Inputinformationen

Satz 130-140 : Löschen des Bildschirms und Ausgabe einer Tabellenüberschrift

Satz 150 : Vorgabe weiterer Ausgangsparameter

Satz 160-170 : Berechnungen für die folgende Periode

Satz 175 : Wenn die neu berechnete Population kleiner oder gleich null wird, ist eine

entsprechende Meldung auszugeben und Sprung zum Satz 21ø

Satz 18ø : Berechnung von Z

Satz 19ø-195 : Ausgabe der Periodenergebnisse

Satz 2øø : Erhöhung des Laufindex I

solange I kleiner als T bleibt, wird Xø mit X1 und Bø mit B1 belegt und das Programm kehrt zurück zum Satz 16ø

Satz 21ø-23ø : Abfrage, ob eine weitere Berechnung gewünscht wird (wenn ja, Löschen des Bildschirms und zurück zu Satz 8ø) und Beendigung des Programms

(7.) SCHRITT : Ergebnisse

Geben wir als Ausgangsbestand z.B. den Wert 1øøø, als Zeithorizont 8 Jahre, als jährliche Wachstumsrate 5 % und als Faktor der Umweltbelastung z.B. den Wert ø.ø1 ein, so erhalten wir die folgenden Angaben :

ZEIT	BESTAND	ZUWACHS %	BELASTUNG
1	1ø5ø	5	15
2	945	-1ø	3ø.75
3	7ø1.66	-25.75	44.93
4	421.52	-39.93	55.45
5	2ø8.87	-5ø.45	61.77
6	9ø.29	-56.77	64.91
7	36.2ø	-59.9	66.26
8	14.ø2	-61.26	66.8ø

NOCHMAL (J/N) ?

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.

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.

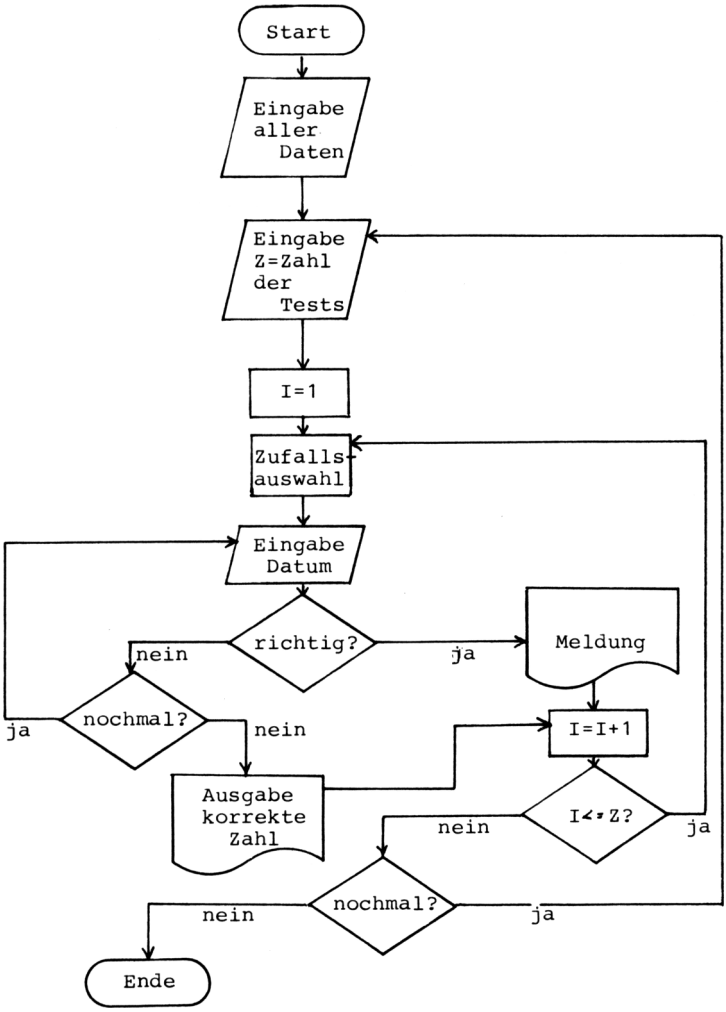
Gibt der Benutzer die korrekte Antwort, so reagiert der Rechner mit Lob; erhält er hingegen keine korrekte Antwort, so soll er den Benutzer fragen, ob dieser noch einen Versuch wagen möchte. Wird dies abgelehnt, soll das Programm dafür sorgen, daß die korrekte Antwort auf dem Bildschirm erscheint.

(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 CLS
30 PRINT"PROGRAMM ZUM ABFRAGEN HISTORISCHER DATEN"
40 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
50 PRINT:PRINT:PRINT:PRINT
60 PRINT"DIESES PROGRAMM BEZIEHT SICH NUR AUF 10"
70 PRINT"AUSGEWAEHLTE HISTORISCHE DATEN (SIEHE"
80 PRINT TAB(5)"DATA-STATEMENTS IN 500 FF.).":PRIN
T
90 PRINT"SOLLEN MEHR DATEN VERWENDET WERDEN, SO"
100 PRINT"MUESSEN DIE DATA ERGAENZT UND N IN SATZ"
110 PRINT TAB(7)"150 VERAENDERT WERDEN."
120 LOCATE 5,23
130 PRINT"BITTE EINE TASTE DRUECKEN !"
140 A$=INKEY$:IF A$="" THEN 140
150 N=10
155 DIM J$(N),S$(N)
160 CLS
170 FOR I=1 TO N:READ J$(I),S$(I):NEXT I
180 PRINT"WIEVIELE ABFRAGEN WERDEN GEWUENSCHT ?"
190 PRINT:INPUT"BITTE ZAHL ANGEBEN : ";Z
200 FOR I=1 TO Z
202 R=INT(RND(1)*N+1)
205 CLS
210 PRINT:PRINT"WANN WAR : ":PRINT
230 PRINT S$(R);:PRINT
240 PRINT:PRINT"IM JAHR : ";:INPUT K$
250 IF K$=J$(R) THEN PRINT:PRINT TAB(9)"R I C H T
:PRINT:GOTO 300
260 PRINT:PRINT"DAS WAR LEIDER NICHT RICHTIG !":PR
INT
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 FOR II = 1 TO 2000:NEXT II: NEXT I
310 PRINT:PRINT:PRINT"NOCH EIN TEST ?";:INPUT" (J/
N) ";A$
320 IF A$="J" THEN CLS: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 DEUTSCH
LAND
530 DATA 9,SCHLACHT IM TEUTOBURGER WALD
540 DATA 1832,HAMBACHER FEST
```

```
550 DATA 1871,GRUENDUNG DES DEUTSCHEN KAISERREICHS
560 DATA 1890,ERFINDUNG DER LOCHKARTE
570 DATA 1812,NAPOLEON VOR MOSKAU
580 DATA 1763,FRIEDE VON HUBERTUSBURG
590 DATA 1495,REICHSTAG VON WORMS
```

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 1ø-12ø : Überschrift und Erläuterungen

Satz 13ø-14ø : Warten

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

Satz 18ø-19ø : Anforderung der Zahl der Tests

Satz 2øø-3øø : Tests :

Satz 2ø2-24ø : Zufallsauswahl eines historischen Ereignisses und Anforderung der Jahreszahl als Antwort des Benutzers

Satz 25ø : Falls korrekt, Meldung und weiter bei 3øø

Satz 26ø-28ø : Falls nicht korrekt, Meldung und Frage, ob noch ein Versuch (wenn ja, zurück nach 2ø5)

Satz 29ø : Wenn nein, korrekte Ausgabe

Satz 3øø : Warteschleife und dann zurück nach Satz 2øø

Satz 31ø-33ø : Frage, ob noch ein Testlauf gewünscht;

wenn ja, zurück zu Satz 18ø nach Löschen des Bildschirms

wenn nein, Ende des Programms

Satz 5øø-59ø : 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 Wünschen erweitert.

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.

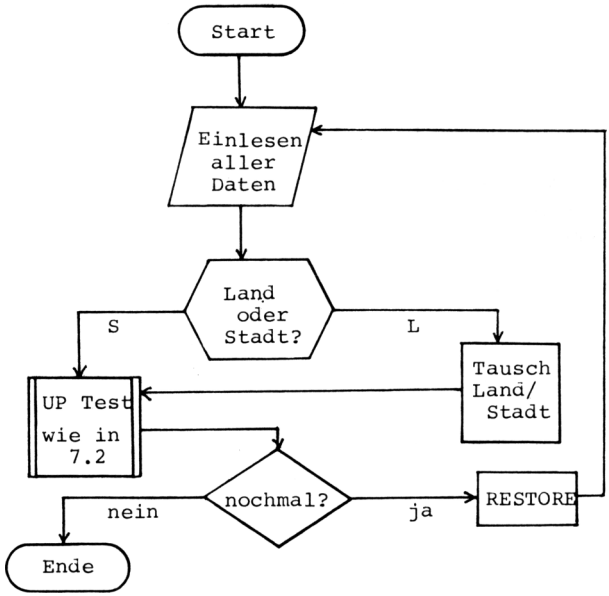
Insoweit bieten sich hier also keine neuen Probleme, so daß wir uns hier sehr kurz fassen können. Auch dieses Programm hat also eher exemplarischen Charakter :

Ohne Schwierigkeiten kann der Benutzer andere Sachverhalte in den DATA-Statements dieses Programms erfasssen 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



(4.) SCHRITT : Programm

```
10 REM E2-LAENDER
20 CLS
30 PRINT"PROGRAMM ZUR LERNKONTROLLE IM ERDKUNDE-"
40 PRINT TAB(14)"UNTERRICHT."
50 PRINT:PRINT:PRINT TAB(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 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !"
150 A$=INKEY$:IF A$="" THEN 150
160 N=11
170 DIM L$(N),S$(N)
180 FOR I=1 TO N:READ L$(I),S$(I):NEXT I
190 CLS
200 PRINT"SOLLEN LAENDER ODER HAUPTSTAEDTE VORGE-"
210 INPUT"GEBEN WERDEN (L/S) ";A$
215 C$="HAUPTSTADT"
220 IF A$="S" THEN C$="LAND":GOTO 250
230 GOSUB 1000:REM TEST
240 GOTO 270
250 FOR I=1 TO N:H$=L$(I):L$(I)=S$(I):S$(I)=H$:NEX
T I
260 GOSUB 1000:REM TEST
270 PRINT:PRINT:INPUT "NOCHMAL ? (J/N) ";A$
280 IF A$="J" THEN CLS:RESTORE:GOTO 180
290 PRINT:PRINT"ENDE":END
```

```
500 DATA SCHLESWIG-HOLSTEIN,KIEL,HAMBURG,HAMBURG,B
REMEN,BREMEN
510 DATA BERLIN,BERLIN,NIEDERSACHSEN,HANNOVER
520 DATA NORDRHEIN-WESTFALEN,DUESSELDORF,HESSEN,WI
ESBADEN
530 DATA RHEINLAND-PFALZ,MAINZ,SAARLAND,SAARBRUECK
EN,BAYERN,MUENCHEN
540 DATA BADEN-WUERTTEMBERG,STUTTGART
1000 REM UP TEST
1010 CLS
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:PRINT TAB(10)"FEHLER !":PRINT
1080 PRINT:PRINT"DIE RICHTIGE ANTWORT LAUTET : "
1090 PRINT:PRINT TAB(10)S$(R):PRINT
1100 NEXT I
1110 RETURN
```

5. SCHRITT : Variablenliste

A$ = Stringvariable
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 10-130 : Überschrift und Erläuterungen

Satz 140-150 : Warten

Satz 160-180 : Einlesen des gesamten Datenbestands

Satz 190-210 : Abfrage, ob Länder oder Hauptstädte vorgegeben werden sollen

Satz 220 : werden Städte gewählt, so wird die Hilfsvariable C$ mit "Land" belegt und es erfolgt ein Sprung nach 250

Satz 230 : Sprung ins Unterprogramm 1000, in dem der eigentliche Abfragetest durchgeführt wird

Satz 24ø : Sprung zum Satz 27ø.

Satz 25ø : Austausch von Ländern und Städten, damit bei anderer Wahl in 19ø-21ø trotzdem wieder das gleiche Unterprogramm verwendet werden kann

Satz 26ø : Wie Satz 23ø

Satz 27ø-29ø : Abfrage, ob noch eine weitere Testserie gewünscht wird (wenn ja, Löschen des Bildschirms und zurück zu Satz 18ø, nachdem der Datenbestand restauriert wurde - dies ist notwendig, um den eventuellen Tausch in 25ø wieder rückgängig zu machen, um also wieder die Ausgangsbedingungen herzustellen); andernfalls Ende des Programms

Satz 5øø-54ø : Ausgangsdaten

Satz 1øøø-111ø : Unterprogramm Test :

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

(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.

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 2ØØØ 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.

(2.) SCHRITT : Problemanalyse

Die geplanten Bevölkerungsprognosen machen es erforderlich, daß man sich für eine bestimmte Prognosemethode entscheidet.

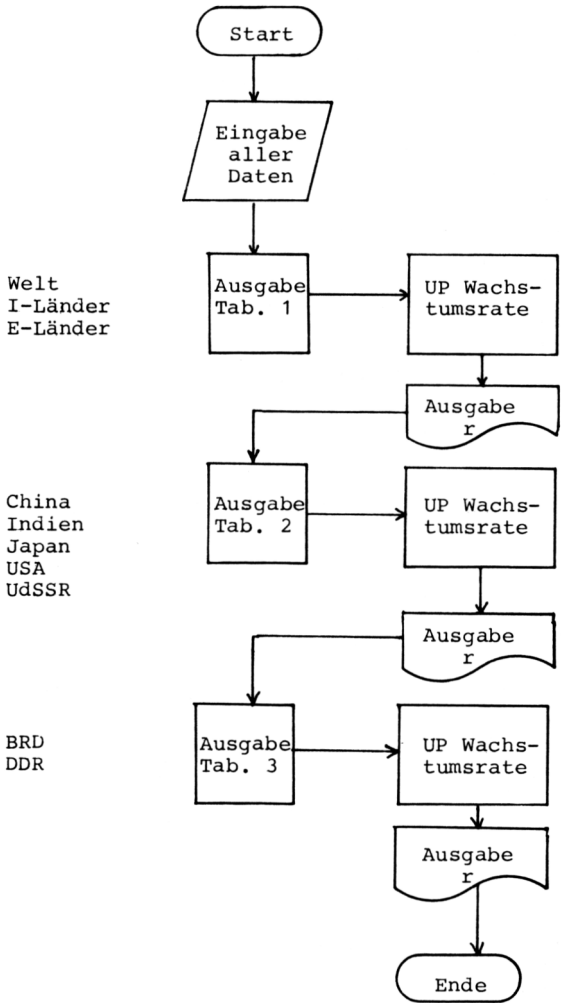
Es gibt eine große Zahl verschiedener Methoden, die im einzelnen nicht hier diskutiert werden können.

Wir haben für das folgende Programm die Werte übernommen die sich bei den Modellprognosen ergeben haben, wie sie in dem Bericht GLOBAL 2ØØØ (Washington 198Ø) 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



(4.) SCHRITT : Programm

```
10 REM E4-BEVOELKERUNG
20 CLS
30 PRINT"PROGRAMM ZUR TABELLARISCHEN DARSTELLUNG"
40 PRINT"DER BEVOELKERUNGSENTWICKLUNG VERSCHIE-"
50 PRINT TAB(12)"DENER LAENDER."
60 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
70 DIM B(10,6),J(6),L$(10),WR(10)
80 FOR I=1 TO 6:READ J(I):NEXT I
90 FOR L=1 TO 10:FOR I=1 TO 6:READ B(L,I):NEXT I:N
EXT L
100 FOR L=1 TO 10:READ L$(L):NEXT L
110 GOSUB 1000 : REM WARTEN
140 PRINT"JAHR    WELT    I-LAENDER    E-LAENDER"
145 PRINT
150 GOSUB 1200:REM STRICH
160 PRINT
170 FOR I=1 TO 6
180 PRINT J(I);TAB(8)B(1,I);TAB(18)B(2,I);TAB(32)B
(3,I):PRINT
190 NEXT I
195 GOSUB 1200:REM STRICH
200 FOR L=1 TO 3
210 X=B(L,1):Y=B(L,6)
220 GOSUB 1100:REM WACHSTUMSRATE
230 NEXT L
240 PRINT:PRINT"RATE %";:PRINT TAB(8)WR(1);TAB(18)
WR(2);TAB(32)WR(3):PRINT
250 PRINT:GOSUB 1000:REM WARTEN
260 CLS
270 PRINT"JAHR  CHINA  INDIEN  JAPAN  USA  UDSSR"
:PRINT
280 GOSUB 1200:REM STRICH
290 FOR I=1 TO 6
300 PRINT J(I);TAB(8)B(4,I);TAB(14)B(5,I);TAB(21)B
(6,I);
302 PRINT TAB(28)B(7,I);TAB(34)B(8,I):PRINT
310 NEXT I
320 GOSUB 1200:REM STRICH
330 FOR L=4 TO 8
340 X=B(L,1):Y=B(L,6)
350 GOSUB 1100:REM WACHSTUMSRATE
360 NEXT L
370 PRINT:PRINT"%";
380 PRINT TAB(8)WR(4);TAB(14)WR(5);TAB(21)WR(6);TA
B(28)WR(7);TAB(34)WR(8)
385 PRINT:PRINT
390 GOSUB 1000:REM WARTEN
```

```
400 CLS
410 PRINT"JAHR  BUNDESREPUBLIK      DDR":PRINT
420 GOSUB 1200:REM STRICH
430 FOR I=1 TO 6
440 PRINT J(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 WACHSTUMSRATE
492 NEXT L
494 PRINT"%";:PRINT TAB(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","I
NDIEN"
620 DATA "JAPAN","USA","UDSSR","BRD","DDR"
1000 REM UP WARTEN
1010 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

```
1100 REM UP WACHSTUMSRATE
1110 R=EXP(LOG(Y/X)/25)-1
1120 R=R*100:R=INT(R*100+0.5)/100
1130 WR(L)=R
1140 RETURN
1200 REM UP STRICH
1210 FOR I=1 TO 40:PRINT"-";:NEXT I
1220 PRINT:RETURN
```

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 Zuwachsraten für die Länder L=1,L=2 und L=3

Satz 21ø : Übernahme des Anfangs- und des Endwertes

Satz 22ø : Sprung ins UP 11øø : Berechnung der Wachstumsrate

Satz 24ø : Ausgabe der Wachstumsraten

Satz 25ø : Warten

Satz 26ø-39ø : Gleiche Vorgehensweise wie in Satz 14ø -25ø beschrieben, nun aber für die zweite Tabelle

Satz 4øø-494 : Entsprechend für die dritte Tabelle

Satz 496 : Beendigung des Programms

Satz 5øø-62ø : Daten

Satz 1øøø-1ø4ø: Unterprogramm Warten

Satz 11øø-114ø: Unterprogramm jahresdurchschnittliche Wachstumsrate (sie wird mit Hilfe der Logarithmenrechnung bestimmt, siehe Satz 111ø)

Satz 12øø-122ø: Unterprogramm Strich :

Durch Aneinanderfügung von Minuszeichen wird auf dem Bildschirm ein Strich gezeichnet

(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	3ØØ3
198Ø	4549	1174	3375
1985	5Ø13	1224	3789
199Ø	5545	1276	4269
1995	6143	1327	4816
2ØØØ	6798	1377	542Ø
RATE %	2.Ø1	.79	2.39

BITTE EINE TASTE DRUECKEN !

2. Tabelle (Angaben in Millionen)

JAHR	CHINA	INDIEN	JAPAN	USA	UDSSR
1975	978	618	112	214	254
198ø	1ø71	694	117	222	268
1985	1151	786	122	235	282
199ø	1241	894	127	248	296
1995	1348	1ø13	131	26ø	31ø
2øøø	1468	1142	135	27ø	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
198ø	61.7	16.7
1985	6ø	16.6
199ø	58.6	16.4
1995	57.8	16.2
2øøø	56.2	16.1
%	-.38	-.17

ENDE

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.

(2.) SCHRITT : Problemanalyse

Nennen wir den Ausgangsbetrag Kø, den Zinssatz P (%) und die Dauer der Bankeinlage T (Jahre).

Dann gilt nach dem ersten Jahr, daß sich der Betrag X1 ergibt zu :

$$X1 = X\emptyset + X\emptyset * P/1\emptyset\emptyset = X\emptyset * (1 + P/1\emptyset\emptyset)$$

Im zweiten Jahr ergibt sich entsprechend zum Jahresende:

$$X2 = X1 + X1 * P/1\emptyset\emptyset = X1 * (1 + P/1\emptyset\emptyset)$$

$$= X\emptyset * (1 + P/1\emptyset\emptyset) * (1 + P/1\emptyset\emptyset)$$

$$= X\emptyset * (1 + P/1\emptyset\emptyset)^2$$

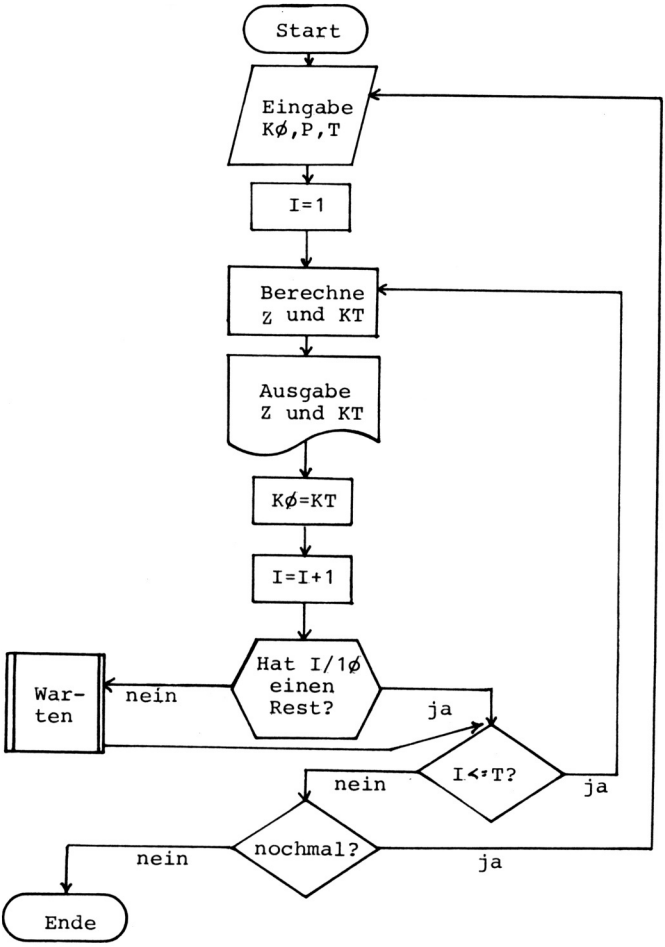
(Diesen Beziehungen sind wir übrigens schon in einem früheren Kapitel begegnet).

Nach T Jahren ergibt sich demnach allgemein :

$$XT = X\emptyset * (1 + P/1\emptyset\emptyset)^T$$

Mit Hilfe dieser Beziehung kann nun ein geeignetes Programm aufgebaut werden.

3. SCHRITT : Flußdiagramm



4. SCHRITT : Programm

```
10 REM W1-ZINSRECHNUNG
20 CLS
30 PRINT"PROGRAMM ZUR BERECHNUNG VON ZINSEN BEI"
40 PRINT"JAEHRLICHER ZINSZAHLUNG UND GEGEBENEM"
50 PRINT TAB(11)"AUSGANGSKAPITAL."
60 PRINT:PRINT:PRINT TAB(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 CLS
120 PRINT"JAHR       ZINS       KAPITAL":PRINT:PRINT
130 FOR I=1 TO T
150 Z=K0*P/100
160 KT=K0+Z
190 PRINT TAB(2)I;:PRINT TAB(8) USING "####.##";Z;
:PRINT TAB(18) USING "######.##";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$="J" THEN CLS:GOTO 80
250 PRINT:PRINT:PRINT"ENDE DER BERECHNUNGEN":END
500 REM UP WARTEN
510 LOCATE 5,23
520 PRINT"BITTE EINE TASTE DRUECKEN !"
530 A$=INKEY$:IF A$="" THEN 530
540 CLS:PRINT"JAHR       ZINS       KAPITAL":PRINT:P
RINT
550 RETURN
```

5. SCHRITT : Variablenliste

I = Laufindex
KØ = 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 1Ø-7Ø : Überschrift

Satz 8Ø-1ØØ : Anforderung der Ausgangsdaten

Satz 11Ø-12Ø : Räumen des Bildschirms und Ausgabe einer Tabellenüberschrift

Satz 13Ø-22Ø : Berechnungen :

15Ø : Zinsbetrag
16Ø : Kapitalbetrag in der nächsten Periode

19Ø : Ausgabe
2ØØ : Neubelegung von KØ mit KT, um in die nächste Periodenberechnung "einsteigen" zu können

21Ø : Nach je 1Ø Perioden Programmunter-

brechung durch Sprung ins UP 5øø

Satz 23ø-25ø : Beendigung des Programms, es sei denn, es wird noch eine Berechnung gewünscht (dann zurück zu Satz 8ø)

Satz 5øø-55ø : Unterprogramm zum Warten, wenn der Bildschirm voll ist :

51ø-52ø : Hinweis
53ø : Warten, bis Tastendruck erfolgt
54ø : Löschen des Bildschirms und erneute Ausgabe der Tabellenüberschrift
55ø : Rücksprung

(7.) SCHRITT : Ergebnisse

Geben wir als Anfangskapitalbestand DM 1øø ein, als Zinssatz 5.5 % und als Zeitdauer 6 Jahre, dann ergibt sich :

JAHR	ZINS	KAPITAL
1	5.5ø	1ø5.5ø
2	5.8ø	111.3ø
3	6.12	117.42
4	6.46	123.88
5	6.81	13ø.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.

2. SCHRITT : Problemanalyse

Zur Erläuterung der Problemlösung gehen wir von folgendem Beispiel aus :

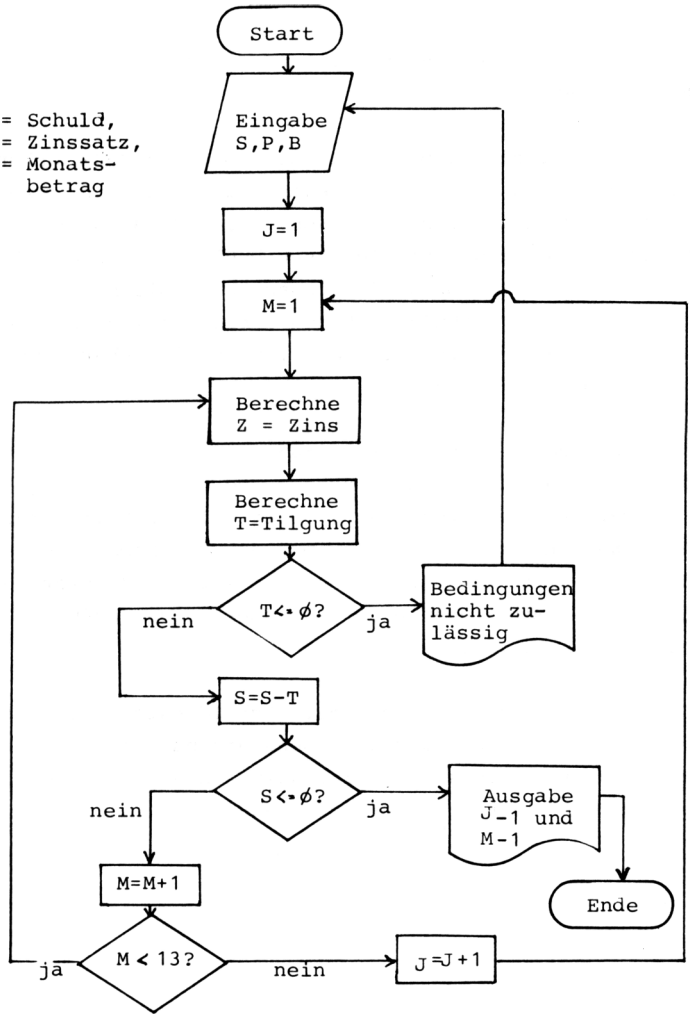
Jemand nimmt eine Hypothek in der Höhe von S DM auf und vereinbart mit der hypothekgewährenden Bank eine monatliche Rückzahlung von B DM. Von diesem Monatsbetrag B müssen zunächst die Schuldzinsen beglichen werden. Der Betrag, der nach Abzug der Zinsen übrig bleibt, kann zur Tilgung der Schuld, d.h. zu ihrer Verringerung verwendet werden - vorausgesetzt, es bleibt überhaupt ein positiver Rest übrig.

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.

Jahr für Jahr soll der aktuelle Schuldenstand ausgegeben werden und wenn der Schuldenstand null oder negativ geworden sein sollte, ist auszugeben, wie lange dieser Schuldentilgungsprozeß gedauert hat.

3. SCHRITT : Flußdiagramm

S = Schuld,
P = Zinssatz,
B = Monats-
betrag



4. SCHRITT : Programm

```
10 REM W2-HYPOTHEK
20 CLS
30 PRINT"PROGRAMM ZUR BERECHNUNG DER DAUER EINER"
40 PRINT TAB(10)"HYPOTHEKENTILGUNG."
50 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT:INPUT"MONATLICHE RATE       : ";B
80 PRINT:INPUT"ZINSSATZ IN %         : ";P
90 PRINT:INPUT"AKTUELLE HYP.SCHULD   : ";S
100 CLS
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+0.5)/100
180 IF S<=0 THEN 300
190 M=M+1:IF M<13 THEN 150
200 PRINT TAB(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"TILGUNGSDAUER : ":PRINT
310 PRINT J-1;" JAHRE UND ";M-1;" MONATE."
320 PRINT:PRINT:PRINT"NOCH EINE BERECHNUNG ? (J/N)
";:INPUT A$
330 IF A$="J" THEN CLS:GOTO 70
340 PRINT:PRINT:PRINT"ENDE DER BERECHNUNG":END
```

5. SCHRITT : Variablenliste

A$ = Stringvariable (Ja, Nein)
B = Monatlicher Rückzahlungsbetrag
I = Laufindex
J = Jahreszähler
M = Monatszähler
P = Jährlicher Schuldzinssatz
T = Monatlicher Tilgungsbetrag
Z = Monatlicher Zinsbetrag

6. SCHRITT : Programmbeschreibung

Satz 1ø-6ø : Überschrift

Satz 7ø-9ø : Vorgabe der Ausgangsinformationen

Satz 1øø-12ø : Ausgabe einer Tabellenüberschrift

Satz 13ø-14ø : Vorgabe eines Jahreszählers (J) und eines Monatszählers M

Satz 15ø-17ø : Berechnung von monatlichem Zins, monatlicher Tilgung und neuer Schuld

Satz 16ø : Wenn nicht getilgt werden kann, dann weiter bei 25ø

Satz 18ø : Ist die Schuld getilgt, dann weiter bei 3øø

Satz 19ø : Nächster Monat
falls M kleiner als 13 ist, weiter bei Satz 15ø, andernfalls bei Satz 2øø

Satz 2øø : Ausgabe der aktuellen Schuld

Satz 21ø : Nächstes Jahr und weiter bei Satz 14ø

Satz 25ø-27ø : Meldung bei unzulässigen Tilgungskonditionen und zurück zu Satz 7ø

Satz 3øø-31ø : Ausgabe der Tilgungsdauer in Jahren und Monaten (der Leser überlege, warum dabei von J und von M jeweils der Wert 1 abgezogen werden muß)

Satz 32ø-34ø : Beendigung des Programms, es sei denn es wird eine erneute Berechnung gewünscht (dann nach Löschen des Bildschirms zurück zum Satz 7ø)

7. SCHRITT : Ergebnisse

Geben wir bei den Anforderungen des Rechners beispielsweise ein :

```
MONATLICHE RATE          :  ? 5øø
ZINSSATZ IN %            :  ? 6.5
AKTUELLE HYP:SCHULD      :  ? 4øøøø
```

so erhalten wir während des Programmablaufs :

```
JAHR      SCHULD
-----------------
  1       36496.86
  2       32759.ø9
  3       28771.ø2
  4       24515.85
  5       199975.7
  6       15131.47
  7       9962.84
  8       4448.ø6

TILGUNGSDAUER :

 8  JAHRE UND  9  MONATE.

NOCH EINE BERECHNUNG   (J/N)  ?
```

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.

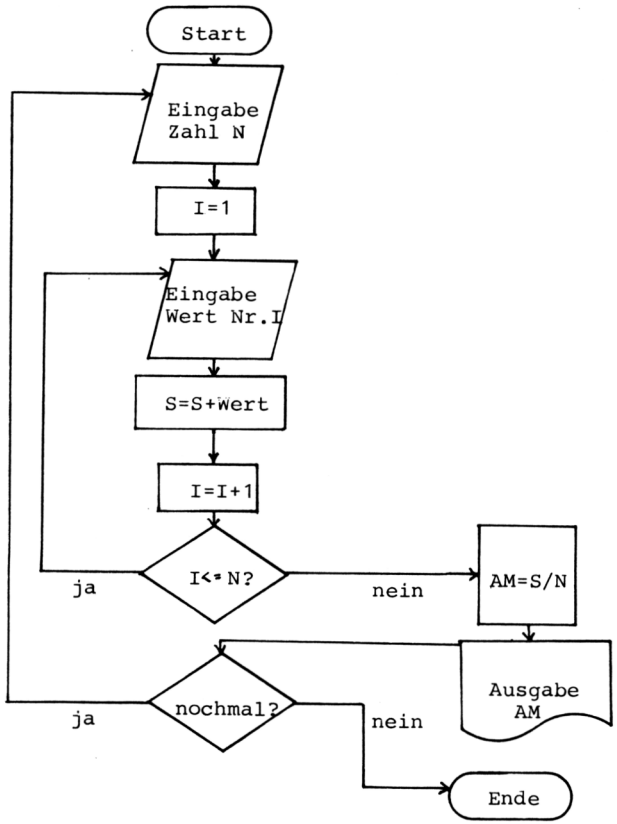
Es muß zunächst dafür gesorgt werden, daß eine offene Anzahl von Daten eingegeben werden kann, weshalb dem Rechner als erstes mitgeteilt werden muß, wieviele Werte es überhaupt sein werden.

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



4. SCHRITT : Programm

```
10 REM W3-MITTELWERT
20 CLS
30 PRINT"PROGRAMM ZUR BERECHNUNG EINES ARITHME-"
40 PRINT TAB(11)"TISCHEN MITTELS."
50 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT:PRINT
70 INPUT"WIEVIELE WERTE : ";N
80 CLS:S=0
90 FOR I=1 TO N
100 PRINT I;". WERT : ";:INPUT X
110 S=S+X
120 NEXT I
130 AM=S/N
150 PRINT:PRINT:PRINT"MITTELWERT = ";AM:PRINT
160 PRINT:PRINT"NOCH EINE BERECHNUNG ? (J/N)";:INP
UT A$
170 IF A$="J" THEN CLS: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

6. SCHRITT : Programmbeschreibung

Satz 1ø-6ø : Überschrift

Satz 7ø : Angabe der Anzahl der zu mittelnden Werte

Satz 8ø : Löschen des Bildschirms und Vorgabe eines Summenfeldes

Satz 9ø-12ø : Eingabe eines Wertes nach dem anderen und Aufaddition aller Werte

Satz 13ø-15ø : Berechnung des arithmetischen Mittels und Ausgabe des Ergebnisses

Satz 16ø-18ø : Abschluß des Programms, es sei denn,es wird eine zusätzliche Berechnung gewünscht

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)  ?
```

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 198ø dient.

Diese Altersverteilung der westdeutschen Bevölkerung stellt sich folgendermaßen dar :

Altersverteilung in der Bundesrepublik Deutschland 198ø :

Altersklasse	Anteil in %
unter 1ø	1ø.3
1ø bis unter 2ø	16.4
2ø bis unter 3ø	14.6
3ø bis unter 4ø	13.4
4ø bis unter 5ø	14.2
5ø bis unter 6ø	11.9
6ø bis unter 7ø	8.9
7ø bis unter 8ø	7.8
8ø bis unter 9ø	2.4
9ø und mehr	ø.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-

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 :

```
1ØØ INPUT "Nächstes Alter : ";X
11Ø IF X < 1Ø THEN Z1=Z1+ 1 : GOTO 1ØØ
12Ø IF X < 2Ø THEN Z2=Z2+ 1 : GOTO 1ØØ
...
usw.

2ØØ R1=(Z1/N)*1ØØ
21Ø R2=(Z2/N)*1ØØ
...
usw.
```

Hier in diesem Beispiel geht es aber lediglich darum, die schon vorgegebenen Werte zunächst tabellarisch darzustellen, was natürlich überhaupt keine programmtechnischen Schwierigkeiten verursacht.

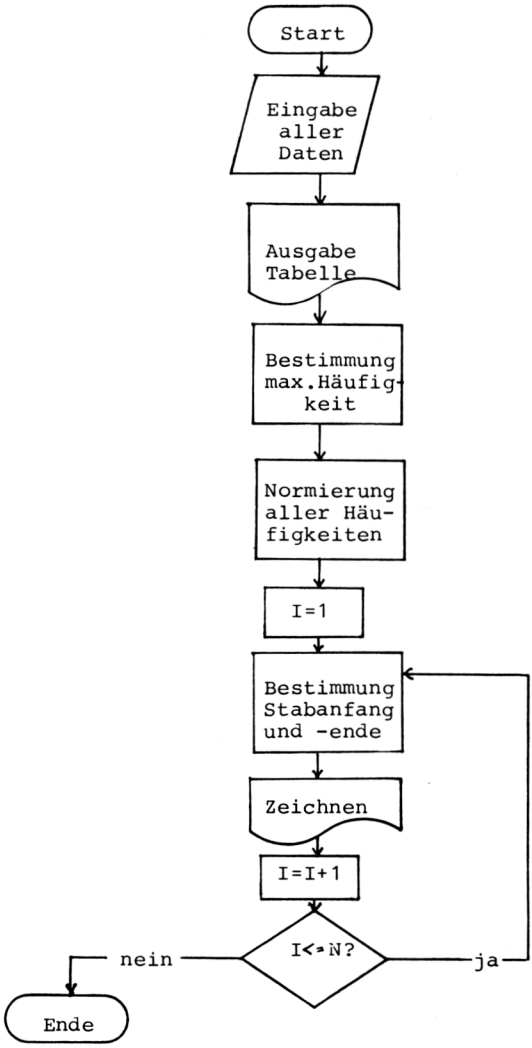
Etwas komplizierter ist hingegen die graphische Darstellung dieser Häufigkeitsverteilung.

Um den Bildschirm für die graphische Darstellung optimal ausnutzen zu können, bestimmen wir zunächst aus dem vorliegenden Datenbestand die größte Häufigkeit und

rechnen alle anderen Häufigkeiten dann so um, daß die größte Häufigkeit in der graphischen Darstellung sich in der Vertikalen über 2ø Bildschirmzeilen erstreckt.

Die Häufigkeiten selbst stellen wir als senkrechte Stäbe dar. Um diese Stäbe maßstabgerecht zeichnen zu lassen, muß ihr jeweiliger Endpunkt bestimmt werden. Zusammen mit dem Anfangspunkt erhalten wir dann jeweils zwei Bildschirmpositionen und es geht dann nur noch darum, alle Positionen zwischen diesen beiden (unter Einschluß derselben) zu belegen.

3. SCHRITT : Flußdiagramm



(4.) SCHRITT : Programm

```
10 REM W5-VERTEILUNG
20 CLS
30 PRINT"PROGRAMM ZUR TABELLARISCHEN UND GRAPHI-"
40 PRINT"SCHEN DARSTELLUNG EINER STATISTISCHEN"
50 PRINT TAB(8)"HAEUFIGKEITSVERTEILUNG."
60 PRINT:PRINT:PRINT TAB(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"
110 GOSUB 1000:REM WARTEN
120 DIM F(10),FT(10)
130 FOR I=1 TO 10:READ F(I):NEXT I
140 PRINT"ALTER";TAB(22)"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
170 PRINT USING "##";A;
172 PRINT TAB(6)" BIS UNTER ";
174 PRINT USING "###";B;
176 PRINT TAB(24) USING "###.#";F(I)
180 NEXT I
190 GOSUB 1000:REM WARTEN
220 REM NORMIERUNG
230 FM=0
240 FOR I=1 TO 10:IF F(I)>FM THEN FM=F(I)
250 NEXT I
260 FOR I=1 TO 10
270 FT(I)=F(I)*(20/FM)
280 NEXT I
290 REM ZEICHNEN
300 B=20
310 FOR I=1 TO 10
320 A=21-FT(I):A=INT(A+0.5)
330 FOR Z=A TO B
340 LOCATE 4*I,Z
350 PRINT CHR$(143)
360 NEXT Z
370 NEXT I
380 PLOT 35,80:DRAW 630,80
```

```
390 PRINT:PRINT TAB(4)"ALTER IN KLASSEN ZU 10 JAHR
EN"
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 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

(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

I = Laufindex

Z = Laufindex (Bildschirmzeile)

(6.) SCHRITT : Programmbeschreibung

Satz 1ø- 9ø : Überschrift

Satz 11ø : Warten

Satz 12ø-13ø : Dimensionieren und Einlesen der Ausgangsdaten

Satz 14ø-15ø : Ausgabe der Tabellenüberschrift

Satz 16ø-18ø : Tabellenausgabe:

165 : Bestimmung von Klassenanfangs- und Klassenendpunkt

17ø-176: Ausgabe

Satz 19ø : Warten

Satz 22ø-25ø : Aufsuchen der größten Häufigkeit durch Vergleich aller Häufigkeiten mit der jeweils schon erreichten größten Häufigkeit (beim Wert null beginnend)

Satz 26ø-28ø : Bestimmung transformierter Häufigkeit durch Ausrichten der Häufigkeiten an der Maximalhäufigkeit

Satz 29ø-37ø : Zeichnen von 1ø Stäben:

- 3øø : Stabende
- 32ø : Bestimmung des Stabanfangs gemessen an der Bildschirmzeileneinteilung
- 33ø-36ø: Zeichnen des Stabes durch Zusammenfügen von kleinen Quadraten in Spalte 4*I

Satz 38ø : Waagrechte Achse

Satz 39ø : Beschriftung der waagrechten Achse

Satz 43ø : Ende des Programms

Satz 5øø : Daten

Satz 1øøø-1ø4ø : Unterprogramm Warten

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 1øøø 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-

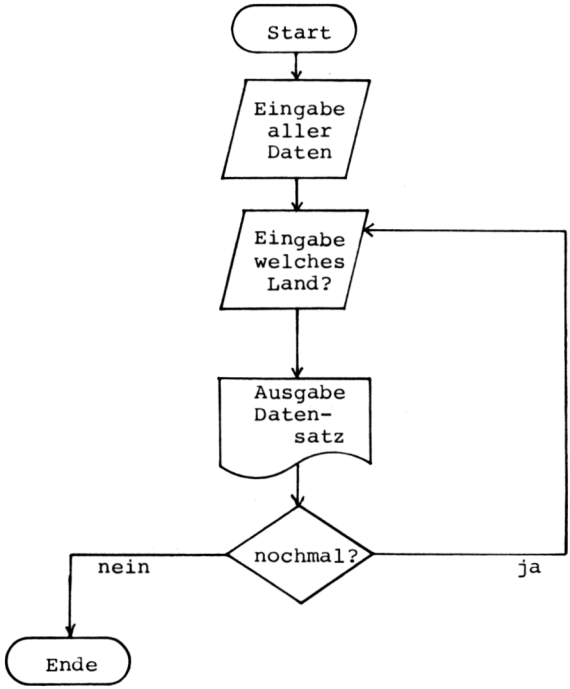
stelle der Bundesländer auch andere Sachverhalte (z.B. Nationen) betrachtet werden könnten. Natürlich müßte dann aber der Datenbestand entsprechend geändert werden.

(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 CLS
30 PRINT"PROGRAMM ZUR AUSGABE VON INFORMATIONEN"
40 PRINT"UEBER DIE WIRTSCHAFTSKRAFT DER BUNDES-"
50 PRINT TAB(16)"LAENDER."
60 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT:PRINT
110 DIM L$(12),V(8,12),B$(8)
120 FOR I=1 TO 12:READ L$(I):NEXT I
130 FOR I=1 TO 8:READ B$(I):NEXT I
140 FOR J=1 TO 8:FOR I=1 TO 12:READ V(J,I):NEXT I:
NEXT J
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$(I)
240 NEXT I
250 PRINT:PRINT:INPUT"BITTE NUMMER ANGEBEN : ";Z
260 CLS
280 PRINT L$(Z):PRINT:PRINT
290 FOR J=1 TO 8
300 PRINT B$(J);TAB(29)": ";:PRINT V(J,Z)
310 NEXT J
320 GOSUB 1000:REM WARTEN
330 PRINT"NOCH EIN ANDERES LAND ? (J/N) ";:INPUT A
$
340 IF A$="J" THEN CLS:GOTO 200
350 PRINT:PRINT"ENDE DER AUSGABE":END
500 DATA "SCHLESWIG-HOLSTEIN","NIEDERSACHSEN","NOR
DRHEIN-WESTFALEN"
502 DATA "HESSEN","RHEINLAND-PFALZ","SAARLAND"
504 DATA "BADEN-WUERTTEMBERG","BAYERN","HAMBURG","
BREMEN","BERLIN"
506 DATA "BUNDESREPUBLIK DEUTSCHLAND"
510 DATA BEVOELKERUNG (MILL.),FLAECHE (1000 QKM),S
OZIALPRODUKT (MRD.DM)
512 DATA ERWERBSTAETIGE (MILL.),AUSLAENDERANTEIL (
%),STEUERAUFKOMMEN (MRD.DM)
514 DATA SCHULDEN (MRD.DM),ANTEIL INDUSTRIEPROD.(%
)
600 DATA 0,0,0,0,0,0,0,0,0,0
602 DATA 0,0,0,0,0,0,0,0,0,0
604 DATA 0,0,0,0,0,0,0,0,0,0
```

```
606 DATA 0,0,0,0,0,0,0,0,0,0
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 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

(5.) SCHRITT : Variablenliste

A$ = Stringvariable
B$ = Variablen-Bezeichnungen
I = Laufindex (Länder)
J = Laufindex (Variablen)
L$ = Ländernamen
V = Variablen
Z = Nummer des gewünschten Landes

(6.) SCHRITT : Programmbeschreibung

Satz 1ø-7ø : Überschrift

Satz 11ø : Dimensionierungen

Satz 12ø-14ø : Einlesen der Ausgangsdaten :

12ø : Ländernamen
13ø : Variablenbezeichnungen
14ø : Variablenwerte

Satz 15ø : Warten

Satz 2øø-25ø : Anforderung der Nummer desjenigen Landes,

über das Informationen gewünscht werden, nachdem eine Länderliste auf dem Bildschirm vorgegeben wurde

Satz 26ø-28ø : Löschen des Bildschirms und Ausgabe des Namens des gewünschten Landes

Satz 29ø-31ø : Ausgabe des Datensatzes für das gewünschte Land

Satz 32ø : Warten

Satz 33ø-35ø : Abfrage, ob noch eine weitere Ausgabe gewünscht wird (wenn ja, Löschen des Bildschirms und zurück zu Satz 2øø); wenn nein, Beendigung des Programms

Satz 5øø-5ø6 : Ländernamen in DATA-Statements.

Satz 51ø-514 : Variablenbezeichnungen in DATA-Statements

Satz 6øø-618 : Variablenwerte :

Die Werte sind so anzugeben, daß für die erste Variable alle Länderwerte, dann für die zweite Variable alle Länderwerte ... usw. folgen (vergl. Satz 14ø)

ACHTUNG : IM PROGRAMM SIND NUR NULLEN (vergl. nächster Arbeitsschritt !)

Satz 1øøø-1ø4ø :Unterprogramm zum Warten.

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 198ø-1982) :

Land	Bevölk. (Mill.)	Fläche (1øøø 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.o	34	4o6	6.9
Hessen	5.6	21	14o	2.4
Rheinland-Pfalz	3.6	2o	81	1.6
Baden-Württem-berg	9.3	36	234	4.3
Bayern	11.o	71	253	5.1
Saarland	1.1	3	24	.4
Hamburg	1.6	.7	75	.7
Bremen	o.7	. .4	22	.3
Berlin	1.9	.5	53	.8
BUND	61.6	248	1484	26.8

Land	Ausländer-anteil	Steuer-aufkommen (Mrd)	Schulden (Mrd)	Anteil Industrieproduktion
Schleswig-Holstein	2.6	10	8	34
Niedersachsen	3.3	31	18	41
Nordrhein-Westfalen	7.1	1o8	32	48
Hessen	7.6	34	13	45
Rheinland-Pfalz	3.6	16	9	44
Baden-Württemberg	1o.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	1o.7	12	7	35
BUND	6.7	365	233	45

Kapitel 9 : Mathematik II

9.1 Vorbemerkung

Abschließend soll ein Kapitel angefügt werden, in dem weitere mathematische Probleme behandelt werden, und zwar solche, die einen etwas höheren Schwierigkeitsgrad aufweisen, als die bisher besprochenen Probleme. Damit soll der bereits fortgeschrittene Mikrocomputer-Benutzer erkennen, 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).

Diese etwas komplizierteren Probleme sollen so programmiert werden, daß keine zusätzlichen BASIC-Anweisungen benötigt werden. Das bedeutet, daß die entstehenden Programme (wie auch schon die vorhergehenden) nicht den Eleganzansprüchen geübter Programmierer entsprechen können. Darauf kommt es aber hier auch nicht an.

Im Gegensatz zu den früheren Kapiteln werden wir uns hier aber kürzer fassen, weil mehr als ein Dutzend zusätzlicher Probleme behandelt werden sollen:

Nach einer kurzen Beschreibung der Problemstellung und einigen Anmerkungen zur Problemanalyse stellen wir sofort das Programm vor und schließen eine knappe Programmbeschreibung an. Insbesondere auf die Arbeitsschritte "Flußdiagramm" und "Ergebnisse" wird also im folgenden verzichtet; nicht nur aus Gründen der Platzersparnis, sondern auch, weil sich unter rein programmlogischen Gesichtspunkten eigentlich keine neuen Detailprobleme mehr stellen.

9.2 Gitter

Problemstellung

Das folgende Programm dient dazu, auf dem Bildschirm ein Liniengitter mit beliebigen Zeilen- und Spaltenabständen auszugeben.
Die Überlegungen, die diesem einfachen Programm zugrundeliegen, können so oder in ähnlicher Weise immer verwendet werden, wenn beispielsweise umfangreichere Tabellenausgaben durch Längs- und Querstriche gegliedert werden sollen.

Problemanalyse

Stichworte zur Problemanalyse sind hier fast entbehrlich: Vorzugeben sind vom Benutzer Angaben zum Zeilen- und zum Spaltenabstand. Entsprechend dieser Angaben wird der Graphik-Cursor per PLOT-Anweisung positioniert, um dann per DRAW-Anweisung eine gerade Linie zu ziehen.

Programm:

```
10 REM Z1-GITTER
20 CLS
30 PRINT"PROGRAMM ZUM ZEICHNEN EINES GITTERS."
40 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
50 LOCATE 3,20:INPUT"SPALTENABSTAND  : ";S
60 LOCATE 3,22:INPUT"ZEILENABSTAND   : ";Z
70 CLS
80 FOR I=1 TO 639 STEP S
90 PLOT I,399:DRAW I,1
100 NEXT I
110 FOR I=1 TO 399 STEP Z
120 PLOT 1,I:DRAW 639,I
130 NEXT I
140 LOCATE 18,24:PRINT"ENDE":END
```

Variablenliste

I = Laufindex
S = Spaltenabstand
Z = Zeilenabstand

Programmbeschreibung:

Satz 1ø- 4ø : Überschrift

Satz 5ø- 6ø : Eingabe von Spalten- und Zeilenabstand

Satz 7ø : Löschen des Bildschirms

Satz 8ø-1øø : Zeichnen der senkrechten Striche

Satz 11ø-13ø : Zeichnen der waagrechten Striche

Satz 14ø : Beendigung des Programms

9.3 Die Gerade

Problemstellung

Das folgende Programm dient dazu, beliebige Geraden in ein Achsenkreuz einzuzeichnen, dessen Ursprung in der Bildschirmmitte (32ø,2øø) liegt.

Problemanalyse

Die Lage einer Geraden wird von zwei Parametern bestimmt, dem Ordinatenabschnitt A und dem Tangens des Steigungswinkels B gemäß der folgenden Funktionalgleichung:

$$Y_i = A + B * X_i$$

Da der Koordinatenursprung im Punkt (32ø,2øø) liegen soll, ergibt sich:

$$Y_i = 2øø + A + B * (X_i - 32ø)$$

Programm:

```
10 REM Z2-GERADE
20 CLS
30 PRINT"PROGRAMM ZUM ZEICHNEN EINER GERADEN."
40 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
50 LOCATE 3,20:INPUT"ORDINATENABSCHNITT A : ";A
60 LOCATE 3,22:INPUT"STEIGUNG           B : ";B
70 CLS
80 PLOT 0,200:DRAW 639,200
90 PLOT 320,399:DRAW 320,0
100 FOR X=0 TO 639
110 Y=200+A+B*(X-320)
120 IF Y<=0 OR Y>=399 THEN 140
130 PLOT X,Y
140 NEXT X
150 LOCATE 18,24:PRINT"ENDE":END
```

Variablenliste

A = Ordinatenabschnitt
B = Tangens des Steigungswinkels
X = Werte auf der X-Achse
Y = Funktionswerte

Programmbeschreibung:

Satz 1ø- 4ø : Überschrift

Satz 5ø- 6ø : Eingabe der Geradenparameter

Satz 7ø : Löschen des Bildschirms

Satz 8ø- 9ø : Zeichnen des Achsenkreuzes

Satz 1øø-14ø : Zeichnen der Gerade über die gesamte Breite des Bildschirms (für unzulässige Y-Werte wird die Zeichenanweisung 13ø übersprungen)

Satz 15ø : Beendigung des Programms

9.4 Der Kreis

Problemstellung

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.

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 :

$$(y - b)^2 = r^2 - (x - a)^2 = D$$

$$y - b = \pm SQR(D)$$

$$y_{1/2} = b \pm SQR(D)$$

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).

Programm:

```
10 REM Z3-KREIS
20 CLS
30 PRINT"PROGRAMM ZUM ZEICHNEN EINES KREISES."
40 PRINT:PRINT:PRINT:PRINT:PRINT
50 PRINT"DIESES PROGRAMM BENOETIGT ALS INPUT-"
60 PRINT"INFORMATIONEN SPALTE UND ZEILE DES "
70 PRINT"KREISMITTELPUNKTES UND RADIUS DES KREI-"
80 PRINT"SES."
90 PRINT:PRINT:PRINT:INPUT"MITTELPUNKT - SPALTE :
";T
100 PRINT:INPUT   "MITTELPUNKT - ZEILE  : ";Z
110 PRINT:PRINT:INPUT "                RADIUS : ";R
120 CLS
130 PLOT 0,0:DRAW 0,399
140 PLOT 0,0:DRAW 639,0
150 PLOT T,0:DRAW T,399
160 PLOT 0,Z:DRAW 639,Z
170 A=T-R:B=T+R
180 IF A<0 THEN A=0
190 IF B>639 THEN B=639
200 FOR X=A TO B
210 D=R*R-(X-T)^2
220 IF D<=0 THEN Y=Z:GOTO 250
230 Y= Z+SQR(D)
240 IF Y<0 OR Y>399 THEN 260
250 PLOT X,Y
260 NEXT X
270 FOR X=B TO A STEP-1
280 D=R*R-(X-T)^2
290 IF D<=0 THEN Y=Z:GOTO 320
300 Y=Z-SQR(D)
310 IF Y<0 OR Y>399 THEN 330
320 PLOT X,Y
330 NEXT X
340 LOCATE 18,24:PRINT"ENDE":END
```

Variablenliste

A = linker Extrempunkt des Kreises
B = rechter Extrempunkt des Kreises
D = D = $R^2-(X-T)^2$
R = Radius
T = Spalte des Kreismittelpunkts
X = X-Koordinatenwerte
Y = Y-Koordinatenwerte
Z = Zeile des Kreismittelpunkts

Programmbeschreibung

Satz 1ø- 8ø : Überschrift und Erläuterungen
Satz 9ø-12ø : Inputinformationen
Satz 13ø-14ø : Achsenkreuz
Satz 15ø-16ø : Hilfs-Achsenkreuz
Satz 17ø-19ø : Festlegung des Wertebereichs für X
Satz 2øø-26ø : Zeichnen eines Halbkreises
Satz 27ø-33ø : Zeichnen des anderen Halbkreises
Satz 34ø : Beendigung des Programms

9.5 Die Sinuslinie

Problemstellung

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.

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.

Programm:

```
10 REM Z4-SINUS
20 CLS
30 PRINT"PROGRAMM ZUM ZEICHNEN EINER SINUSLINIE."
40 PRINT:PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS,
1984"
50 PRINT:PRINT:PRINT:PRINT
60 PRINT"DAS PROGRAMM BENOETIGT DIE FOLGENDEN"
70 PRINT"INPUTINFORMATIONEN :":PRINT:PRINT
80 PRINT:INPUT"FREQUENZ (IN EINHEITEN VON PI) ";P
90 PRINT:INPUT"AMPLITUDE (0 BIS 199)          ";A
100 CLS
110 PLOT 0,200:DRAW 639,200
120 FOR X=0 TO 639
130 J=X/100:J=J*(1/P)
140 Y=SIN(J):Y=Y*A
150 Y=200+Y
160 PLOT X,Y
170 NEXT X
172 LOCATE 19,12:PRINT"PI"
174 LOCATE 38,12:PRINT"2PI"
176 LOCATE 3,21:PRINT"FREQUENZ (IN PI) = ";P
178 LOCATE 3,22:PRINT"AMPLITUDE        - ";A
180 LOCATE 18,24:PRINT"ENDE":END
```

Variablenliste :

A = Amplitude
J = transformierter X-Wert
P = Frequenz
X = X-Koordinatenwert
Y = Y-Koordinatenwert

Programmbeschreibung

Satz 1ø- 5ø : Überschrift
Satz 6ø- 9ø : Inputinformationen
Satz 1øø-11ø : Zeichnen einer waagrechten Mittellinie
Satz 12ø-17ø : Bestimmung der Sinuswerte und Zeichnen nach Transformation gemäß P und A
Satz 172-18ø : Textausgabe und Beendigung des Programms

9.6 Mathem. Tafel 1

Problemstellung

Im folgenden Programm werden für alle natürlichen Zahlen von 1 bis 1øø die Quadratzahlen und die Quadratwurzeln ausgegeben.
Insoweit ist dieses Programm quasi als "Nachschlageprogramm" zu verstehen.

Problemanalyse

Eine Problemanalyse dürfte bei dieser sehr einfachen Problemstellung entbehrlich sein.

Programm:

```
10 REM Z5-TAFEL1
20 CLS
30 PRINT"PROGRAMM ZUR ERZEUGUNG DER QUADRATZAH-"
40 PRINT"LEN UND DER QUADRATWURZELN FUER ALLE"
45 PRINT TAB(7)"NATUERLICHEN ZAHLEN (1-100)."
50 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
60 GOSUB 2000:REM WARTEN
70 FOR I=1 TO 100
80 Q=I*I:W=SQR(I)
90 PRINT USING "###";I;:PRINT TAB(7) USING "#####"
;Q;:PRINT TAB(16) USING "###.###";W
100 IF I/15=INT(I/15) THEN GOSUB 2000:REM WARTEN
110 NEXT I
120 PRINT:PRINT"ENDE DER AUSGABE":END
1000 REM UP KOPFZEILE
1010 PRINT"ZAHL   QUADRAT   WURZEL":PRINT
1020 FOR J=1 TO 23:PRINT"-";:NEXT J:PRINT:RETURN
2000 REM UP WARTEN
2010 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !
2020 A$=INKEY$:IF A$="" THEN 2020
2030 CLS
2040 GOSUB 1000: REM KOPFZEILE
2050 RETURN
```

Variablenliste:

A$ = Stringvariable für Tasteneingabe
I = Laufindex (natürliche Zahlen 1-1ØØ)
J = Laufindex
Q = Quadratzahl
W = Quadratwurzel

Programmbeschreibung

Satz 1Ø - 5Ø : Überschrift

Satz 6Ø : Sprung ins Unterprogramm 2ØØØ zum Abwarten

Satz 7Ø -11Ø : Berechnung und Ausgabe der interessierenden Werte

Satz 1ØØ: Nach je 15 Zeilen Sprung ins Unterprogramm 2ØØØ zum Abwarten

Satz 12Ø : Ende des Hauptprogramms

Satz 1ØØØ-1Ø2Ø : Unterprogramm zur Ausgabe der Tabellen-Kopfzeile

---

Satz 2ØØØ-2Ø5Ø : Unterprogramm zum Abwarten

Satz 2Ø1Ø: Ausgabe einer Meldung

Satz 2Ø2Ø: Abwarten einer Tasteneingabe

Satz 2Ø3Ø-2Ø5Ø:

Ist die Tasteneingabe erfolgt, Löschen des Bildschirms, Ausgabe einer neuen Kopfzeile durch Sprung ins UP 1ØØØ und danach Rückkehr ins Hauptprogramm.

9.7 Mathem. Tafel 2

Problemstellung

Dieses Programm dient zur tabellarischen Ausgabe der Winkelfunktionen

- Sinus (SIN)
- Cosinus (COS)
- Tangens (TAN)

Diese Winkelfunktionen werden in Abhängigkeit von Einheiten des Kreisparameters π (im Programm Pi genannt) ausgegeben. Deshalb werden in der sich ergebenden Tabelle auch Winkelgrade angegeben.

Problemanalyse

Auch hier ist eine Problemanalyse nicht erforderlich, weil keine besonderen Schwierigkeiten auftauchen.

Programm:

```
10 REM Z6-TAFEL2
20 CLS
30 PRINT TAB(11)"ZAHLENTAFEL  2":PRINT:PRINT
40 PRINT"PROGRAMM ZUR BESTIMMUNG DER WERTE DER"
50 PRINT"WINKELFUNKTIONEN :":PRINT
60 PRINT TAB(11)"SINUS      (SIN)"
70 PRINT TAB(11)"COSINUS    (COS)"
80 PRINT TAB(11)"TANGENS    (TAN)"
85 PRINT:PRINT
90 GOSUB 2000:REM WARTEN
100 FOR J=0 TO 40
102 I=J/10
105 A=(360*I)/(2*(22/7)):A=INT(A+0.5)
110 B=SIN(I)
120 C=COS(I)
130 D=TAN(I)
135 PRINT USING "##.#";I;:PRINT TAB(6) USING "###.
###";A;
136 PRINT TAB(14) USING "###.###";B;:PRINT TAB(21)
 USING "###.###";C;:PRINT TAB(28) USING "###.###";
D
137 IF J=0 THEN 150
140 IF J/10=INT(J/10) THEN GOSUB 2000
150 NEXT J
160 PRINT:PRINT"ENDE DER AUSGABE":END
1000 REM UP KOPFZEILE
1010 PRINT:PRINT" PI    GRAD    SIN    COS    TAN"
:PRINT
1020 RETURN
2000 REM UP WARTEN
2010 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !
"
2020 A$=INKEY$:IF A$="" THEN 2020
2030 CLS
2040 GOSUB 1000
2050 RETURN
```

Variablenliste:

A = Winkel (in Grad)
A$ = Stringvariable für Tasteneingabe
B = Sinus
C = Cosinus
D = Tangens
I = J/1ø (=Abszissenwert der Winkelfunktionen)
J = Laufindex (hier ø-4ø)

Programmbeschreibung:

Satz 1ø - 85 : Überschrift

Satz 95 : Sprung ins Unterprogramm 2øøø zum Abwarten

Satz 1øø-15ø : Berechnungen und Ausgabe der Ergebnisse.

Es werden 41 Tabellenwerte bestimmt (J=ø TO 4ø), wobei die Winkelfunktionen auf I=J/1ø angewandt werden (Satz 1ø2).

Satz 1ø5: Umwandlung in Winkelgrad

Satz 11ø-13ø: Bestimmung von sin, cos und tan

Satz 135-137: Ausgabe

Satz 14ø: Sprung ins Unterprogramm 2øøø nach jeder 1ø. Runde. (Um bei J=ø diesen Sprung zu vermeiden, der dann noch nicht sinnvoll ist, wird in Satz 137 für J=ø zum Satz 15ø gesprungen).

Satz 16Ø	: Ende des Hauptprogramms
Satz 1ØØØ-1Ø2Ø	: Unterprogramm zur Ausgabe der Tabellen-Kopfzeile
Satz 2ØØØ-2Ø5Ø	: Unterprogramm zum Warten (Beschreibung siehe Programm Z6-TAFEL1).

9.8 Mathem. Tafel 3

Problemstellung

Das folgende Programm dient zur Ausgabe der natürlichen Logarithmen ln(X) im Wertebereich für X zwischen 1 und 1øø.

Problemanalyse

Auch hier ist eine Problemanalyse wegen der Schlichtheit der Fragestellung entbehrlich.

Programm:

```
10 REM Z7-TAFEL3
20 CLS
30 PRINT TAB(13)"ZAHLENTAFEL  3":PRINT:PRINT
40 PRINT"PROGRAMM ZUR BESTIMMUNG DER WERTE DER"
50 PRINT TAB(8)"NATUERLICHEN LOGARITHMEN"
60 PRINT:PRINT:PRINT TAB(10)"PROF.DR.W.VOSS, 1984"
:PRINT
70 GOSUB 2000
80 FOR I=1 TO 100
85 L=LOG(I)
90 PRINT USING "####";I;:PRINT TAB(8) USING "###.#
###";L
100 IF I/15=INT(I/15) THEN GOSUB 2000
110 NEXT I
120 PRINT:PRINT:PRINT"ENDE DER AUSGABE":END
1000 REM UP KOPFZEILE
1010 PRINT:PRINT"ZAHL   LOGARITHMUS":PRINT
1020 RETURN
2000 REM UP WARTEN
2010 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !
"
2020 A$=INKEY$:IF A$="" THEN 2020
2030 CLS
2040 GOSUB 1000
2050 RETURN
```

Variablenliste:

A$ = Stringvariable für Tasteneingabe
I = Laufindex (hier 1-1ØØ)
L = Natürlicher Logarithmus von I

Programmbeschreibung:

Satz 1Ø - 6Ø : Überschrift

Satz 7Ø : Sprung ins Unterprogramm 2ØØØ zum Abwarten

Satz 8Ø -11Ø : Berechnungen der Logarithmen und Ausgabe;
nach jeder 15. Tabellenzeile Sprung ins Unterprogramm 2ØØØ zum Warten

Satz 12Ø : Beendigung des Hauptprogramms

Satz 1ØØØ-1Ø2Ø : Unterprogramm zur Ausgabe einer Tabellen-Kopfzeile

---

Satz 2ØØØ-2Ø5Ø ; Unterprogramm zum Abwarten (Beschreibung siehe Programm 36-TAFEL1).

9.9 Ellipse

Problemstellung

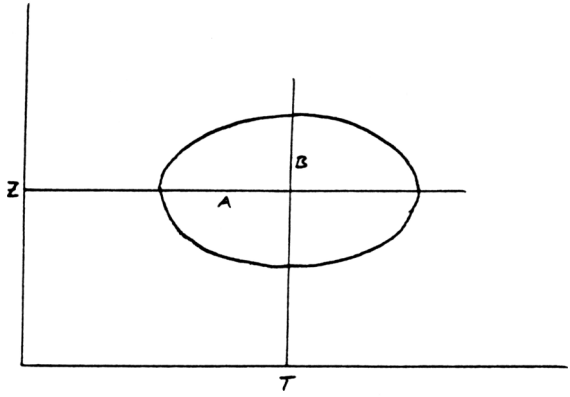
Das Programm zeichnet eine beliebige Ellipse.
Dabei sind folgende Werte einzugeben:

T: Spalte des Ellipsenmittelpunkts
(Ø-319)

Z: Zeile des Ellipsenmittelpunkts
(Ø-199)

A: Erste Halbachse der Ellipse

B: Zweite Halbachse der Ellipse



Problemanalyse

Die Ellipsengleichung lautet in der sog. Mittelpunkt-Schreibweise (Ellipsenmittelpunkt im Koordinatenursprung):

$$\frac{x^2}{a} + \frac{y^2}{b} = 1$$

Um eine derartige Funktion per BASIC-Programm zeichnen zu können, müssen wir sie nach y auflösen, wobei sich ergibt:

$$y_{1/2} = \pm \frac{b}{a} \sqrt{a^2 - x^2}$$

Wenn wir nun berücksichtigen, daß eine beliebige Ellipse die Koordinaten (T,Z) und nicht immer nur (Ø,Ø) als Mittelpunktkoordinaten aufweist, so sind Verschiebungskorrekturen erforderlich (vergl. die folgende Programmbeschreibung).

Programm:

```
10 REM Z8-ELLIPSE
20 CLS
30 PRINT"PROGRAMM ZUM ZEICHNEN EINER BELIEBIGEN"
40 PRINT TAB(16)"ELLIPSE."
50 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT"DIESES PROGRAMM BENOETIGT ALS INPUT-"
80 PRINT"INFORMATIONEN :":PRINT
90 PRINT"KOORDINATEN DES MITTELPUNKTS : ":PRINT
100 PRINT TAB(5);:INPUT"X-WERT (SPALTE)  : ";T
110 PRINT TAB(5);:INPUT"Y-WERT (ZEILE)   : ";Z
120 PRINT:INPUT"ERSTE HALBACHSE A   : ";A
125 PRINT:INPUT"ZWEITE HALBACHSE B  : ";B
130 CLS
140 PLOT 0,0:DRAW 639,0
150 PLOT 0,0:DRAW 0,399
160 PLOT 0,Z:DRAW 639,Z
170 PLOT T,0:DRAW T,399
390 REM ELLIPSE
400 C=T-A:D=T+A
410 IF C<0 THEN C=0
420 IF D>639 THEN D=639
430 FOR X=C+1 TO D-1
440 DS=(B/A)*SQR(A*A-(X-T)^2)
450 Y=Z+DS
460 IF Y<0 OR Y>399 THEN 480
470 PLOT X,Y
480 NEXT X
490 FOR X=D-1 TO C+1 STEP -1
500 DS=(B/A)*SQR(A*A-(X-T)^2)
510 Y=Z-DS
520 IF Y<0 OR Y>399 THEN 540
530 PLOT X,Y
540 NEXT X
550 LOCATE 5,23:PRINT"ENDE DER AUSGABE":END
```

Variablenliste:

A = Erste Halbachse der Ellipse
B = Zweite Halbachse der Ellipse
C = Linker Extremwert der Ellipse
D = Rechter Extremwert der Ellipse
DS = Hilfsgröße aus der Ellipsengleichung
T = Spalte des Ellipsenmittelpunkts
X = Abszissenwert
Y = Ordinatenwert
Z = Zeile des Ellipsenmittelpunkts

Programmbeschreibung

Satz 1ø- 6ø : Überschrift

Satz 6ø-125 : Vorgabe der Mittelpunktkoordinaten und der Halbachsen

Satz 13ø : Löschen des Bildschirms

Satz 14ø-15ø : Zeichnen des Achsenkreuzes

Satz 16ø-17ø : Zeichnen eines zweiten Achsenkreuzes, dessen Achsen durch den Ellipsenmittelpunkt laufen

Satz 39ø-48ø : Zeichnen der oberen Ellipsenhälfte

Satz 49ø-54ø : Zeichnen der unteren Ellipsenhälfte

Satz 55ø : Beendigung des Programms

9.10 Gauß'sche Normalverteilung

Problemstellung

Dieses Programm stellt die Gauß'sche Normalverteilung graphisch dar.
Sie hat die Dichtefunktion

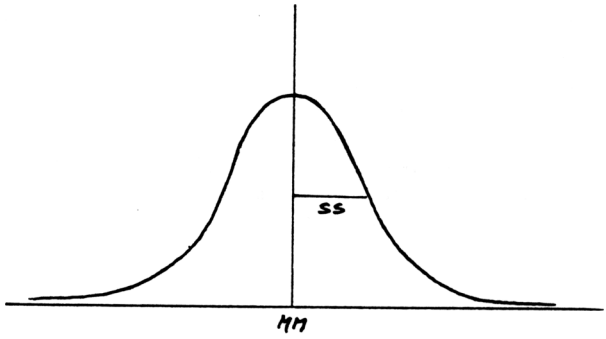
$$f(x) = \frac{1}{SS \cdot \sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{X - MM}{SS}\right)^2}$$

Dabei sind MM und SS die charakterisierenden Lageparameter (siehe folgende Skizze):

- MM bezeichnet den Mittelwert der Verteilung
- SS bezeichnet die Streuung der Verteilung (Standardabweichung = Abstand zwischen dem Lot im Mittelwert und einem der beiden Wendepunkte der Verteilung).

In diesem Programm haben wir vorgegeben:

MM = 160
SS = 40



Problemanalyse

Hier wiederum entbehrlich.

Programm:

```
10 REM Z9-GAUSS
20 CLS
30 PRINT"PROGRAMM ZUM ZEICHNEN EINER GAUSS'SCHEN"
40 PRINT TAB(11)"NORMALVERTEILUNG."
50 PRINT:PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS,
1984"
60 PRINT:PRINT:PRINT:PRINT
70 PRINT"DIESES PROGRAMM ZEICHNET EINE NORMAL-"
80 PRINT"VERTEILUNG MIT DEM MITTELWERT 200 UND"
90 PRINT"DER STANDARDABWEICHUNG 60.":PRINT
100 PRINT"DIESE WERTE KOENNEN GGF. GEAENDERT WER-"
110 PRINT TAB(10)"DEN. (SIEHE SATZ 120)"
115 GOSUB 1000:REM WARTEN
120 MM=200:SS=60
130 CLS
140 PLOT 0,0:DRAW 400,0
150 PLOT 200,0:DRAW 200,300
390 REM NORMALVERTEILUNG
400 A=1/(SS*SQR(2*(22/7)))
410 FOR X=0 TO 400
420 B=((X-MM)/SS)^2
430 B=0.5*B*(-1)
440 Y=A*EXP(B)
450 Y=(2000000/SS)*Y
460 IF Y<0 OR Y>399 THEN 480
470 PLOT X,Y
480 NEXT X
490 PRINT"ENDE DER AUSGABE":END
1000 REM UP WARTEN
1010 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !
"
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS:RETURN
```

Variablenliste:

A = Erster Term der Funktionalgleichung
A$ = Stringvariable für Tasteneingabe
B = Exponent in der Funktionalgleichung
MM = Mittelwert der Normalverteilung
SS = Standardabweichung der Normalverteilung
X = Abszissenwert
Y = Ordinatenwert

Programmbeschreibung

Satz 1ø-11ø : Überschrift und Erläuterungen

Satz 115 : Abwarten

Satz 12ø-13ø : Vorgabe der Funktionalparameter der Verteilung und Löschen des Bildschirms

Satz 14ø-15ø : Zeichnen eines Achsenkreuzes

Satz 39ø-48ø : Errechnen der Ordinatenwerte der Normalverteilung, Normierung (Satz 45ø) und graphische Ausgabe

Satz 49ø : Ende des Programms

Satz 1øøø-1ø3ø : Unterprogramm zum Abwarten

9.11 Regression

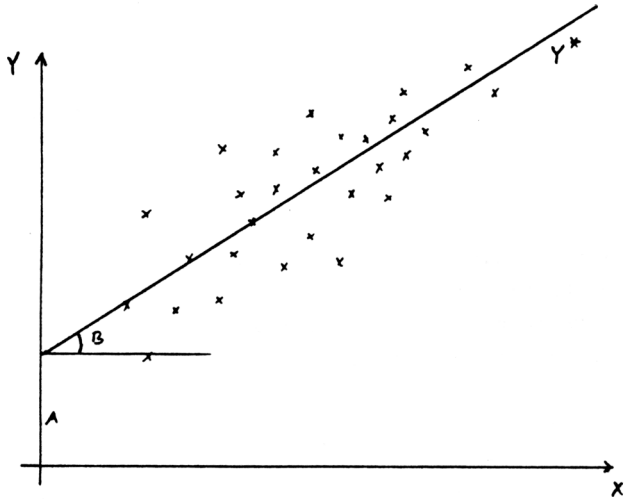
Problemstellung

Die Regressionsrechnung ist eine wichtige Aufgabenstellung aus dem Bereich der mathematischen Statistik. Es geht dabei darum, eine bivariate statistische Verteilung durch eine zusammenfassende mathematische Funktion, die im einfachsten Fall eine Gerade ist, zu beschreiben.

Eine bivariate statistische Verteilung entsteht beispielsweise dann, wenn wir eine gegebene Population nach zwei Merkmalen gleichzeitig untersuchen, wenn wir also zum Beispiel eine Reihe von Untersuchungspersonen nach Körpergröße und Körpergewicht gleichzeitig befragen.

Ziel einer solchen Untersuchung ist es, festzustellen, ob zwischen den beiden Variablen ein Zusammenhang besteht.

Eine solche bivariate Verteilung läßt sich anschaulich als "Punktwolke" in einem zweidimensionalen Koordinatensystem darstellen, wie es die folgende Skizze verdeutlicht.



Um nun die Art des Zusammenhangs zwischen beiden Variablen zu beschreiben, bzw. um bei einer Zeitreihe den linearen Trend zu bestimmen, was mathematisch die gleiche Aufgabe ist, versucht man, in die Punkte im Achsenkreuz möglichst gut eine lineare mathematische Funktion hineinzulegen.

Entsprechend gibt es auch nichtlineare Berechnungsansätze, auf die aber hier nicht eingegangen werden soll.

Problemanalyse

Die einzuzeichnende lineare oder nichtlineare Funktion bestimmt man in der Regel mit der Methode der kleinsten Quadrate.

Diese Methode gibt eine Rechenvorschrift an zur Bestimmung der Parameter der anzupassenden Funktion. Bei einer Geraden handelt es sich dabei um den Ordinatenabschnitt A und die Steigung B.

Die Methode der kleinsten Quadrate verlangt, daß diese beiden Parameter so zu bestimmen sind, daß die Summe der quadrierten Abweichungen zwischen den beobachteten Y-Werten y_i (senkrechte Koordinaten) und den auf der Gerade liegenden Y-Werten y_i^* zu minimieren ist, also:

$$\sum (y_i - y_i^*)^2 = \text{Min.}$$

Da die Werte y_i^* sich gemäß der Geradengleichung wie folgt berechnen :

$$y_i^* = a + b^* x_i$$

ergibt sich aus der obigen Minimierungsregel :

$$\sum(y_i - a - b*x_i)^2 = \text{Min.}$$

Das Minimum dieser Beziehung findet man, indem man sie partiell nach den Größen a und b differenziert und diese beiden ersten partiellen Ableitungen gleich null setzt.

Auf diese Weise erhält man zwei Gleichungen mit zwei Unbekannten (a und b), die man nach eben diesen Unbekannten auflösen kann. Damit erhält man die folgenden beiden Bestimmungsgleichungen für a und b :

$$b = \frac{n * \sum x_i y_i - \sum x_i \sum y_i}{n * \sum x_i^2 - (\sum x_i)^2}$$

$$a = \frac{\sum y_i}{n} - b * \frac{\sum x_i}{n}$$

Das zu entwickelnde BASIC-Programm dient dazu, für beliebig einzugebende Wertepaare x_i, y_i diese beiden Parameter nach der Methode der kleinsten Quadrate zu errechnen und als Ergebnisse auszugeben.

Programm:

```
10 REM Z10-REGRESSION
20 CLS
30 PRINT"PROGRAMM ZUR BESTIMMUNG DER PARAMETER"
40 PRINT"EINER LINEAREN REGRESSIONSFUNKTION BZW."
50 PRINT TAB(6)"EINER LINEAREN TRENDLINIE."
60 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT:PRINT
80 PRINT"DIESES PROGRAMM BENOETIGT WERTEPAARE"
90 PRINT"ALS INPUT-INFORMATIONEN (Z.B. X- UND Y-"
100 PRINT"WERTE ODER ZEIT- UND MERKMALSWERTE USW.)
110 GOSUB 1000 :REM WARTEN
120 INPUT"WIEVIELE WERTEPAARE : ";N
130 DIM X(N),Y(N)
140 PRINT:PRINT TAB(5)"EINGABE : ":PRINT:PRINT
150 FOR I=1 TO N
160 PRINT I;".PAAR X,Y : ";:INPUT X(I),Y(I)
170 NEXT I
172 CLS:PRINT"KONTROLLE":PRINT:PRINT
174 PRINT"NR.          X-WERT        Y-WERT":PRINT
176 FOR I=1 TO N:PRINT I,X(I),Y(I)
178 IF I/10=INT(I/10) THEN GOSUB 1000
179 NEXT I
180 REM REGRESSION
185 GOSUB 1000:REM WARTEN
190 FOR I=1 TO N
200 S1=S1+X(I)*Y(I)
210 S2=S2+X(I):S3=S3+Y(I)
220 S4=S4+X(I)*X(I)
230 NEXT I
240 B=(N*S1-S2*S3)/(N*S4-S2*S2)
250 A=S3/N-B*S2/N
```

```
265 CLS
270 PRINT"ERGEBNIS : ":PRINT:PRINT
280 PRINT"ORDINATENABSCHNITT A = ";A
290 PRINT:PRINT"STEIGUNG            B = ";B
300 PRINT:PRINT:PRINT:PRINT
310 PRINT"MIT DIESEN PARAMETERN KANN NUN DAS PRO-"
320 PRINT"GRAMM ZUM ZEICHNEN EINER GERADEN AUFGE-"
330 PRINT"RUFEN WERDEN, WENN MAN DAS ERGEBNIS "
340 PRINT"AUCH GRAPHISCH VERANSCHAULICHEN MOECHTE.
350 PRINT:PRINT:PRINT"ENDE DER AUSGABE":END
1000 REM UP WARTEN
1010 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

Variablenliste:

A = Ordinatenabschnitt der Geraden
A$ = Stringvariable für Tastatureingabe
B = Steigung der Geraden
I = Laufindex
N = Zahl der Wertepaare
S1 =
S2 = Hilfssummen zur Berechnung von B und A
S3 =
S4 =
X = Abszissenwert
Y = Ordinatenwert

Programmbeschreibung:

Satz 1ø-11ø : Überschrift, Erläuterungen und Warten.

Satz 12ø-13ø : Angabe der Zahl der Paare von Merkmalswerten und Dimensionierung.

Satz 14ø-17ø : Eingabe der Merkmalswerte über INPUT.

Satz 172-179 : Kontrollausgabe der Paare von Merkmalswerten mit Pause nach jedem zehnten Paar.

Satz 18ø-185 : Warten.

Satz 19ø-23ø : Bildung der vier Summen, die in den Berechnungsformeln für A und B benötigt werden.

Satz 24ø-25ø : Berechnung der Geradenparameter A und B.

Satz 265-3øø : Ergebnisausgabe.

Satz 31ø-35ø : Verzweigungshinweis und Beendigung des Programms.

Satz 1øøø-1ø4ø : Unterprogramm Warten.

9.12 Permutationen

Problemstellung

Die Berechnung von Permutationen beantwortet die Frage, auf wieviele unterschiedliche Arten verschiedene Elemente angeordnet werden können.

Problemanalyse

Haben wir beispielsweise drei verschiedene Elemente a, b und c, so sind die folgenden Anordnungsmöglichkeiten voneinander unterscheidbar :

abc, acb, bac, bca, cab, cba

Bei drei Elementen gibt es also sechs verschiedene Anordnungsmöglichkeiten.

Allgemein gilt die folgende Überlegung :

Für das erste von drei Elementen gibt es drei Möglichkeiten : Es kann an erster, an zweiter oder an dritter Stelle stehen;

für das zweite Element gibt es dann nur noch zwei Möglichkeiten, wenn entschieden worden ist, wo das erste Element stehen soll.

Dies bedeutet aber, daß es für die ersten beiden Elemente zusammen 3*2 = 6 Möglichkeiten gibt.

Für das dritte Element gibt es dann nur noch eine Möglichkeit und für alle drei Elemente also 3*2*1 = 6 verschiedene Anordnungsmöglichkeiten.

Hat man nun n verschiedene Elemente, so ergibt sich entsprechend :

Zahl der Anordnungsmöglichkeiten =

n*(n-1)*(n-2)*(n-3)* ... * 3*2*1

Diese Produktkette kürzt man ab durch die Schreibweise n! (sprich n-Fakultät) und sagt, daß n verschiedene Elemente auf n-Fakultät verschiedene Arten angeordnet werden können, oder :

Permutation von n verschiedenen Elementen ist n-Fakultät (n!).
Es ist zu beachten, daß bei großen Werten für n der Ausdruck n! rasch sehr groß wird, und zwar so rasch ansteigend, daß schon ab n = 34 der Rechner das Ergebnis 34! nicht mehr darstellen kann.

Programm

```
10 REM Z11-PERMUTATIONEN
20 CLS
30 PRINT"PROGRAMM ZUR BESTIMMUNG DER ANZAHL DER"
40 PRINT"PERMUTATIONEN VON N VERSCHIEDENEN ELE-"
50 PRINT TAB(5)"MENTEN (= N-FAKULTAET = N!)"
60 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT
80 INPUT"ZAHL DER ELEMENTE N : ";N
90 P=N
100 IF N>=34 THEN PRINT"N IST ZU GROSS FUER DIESEN
 RECHNER":GOTO 150
110 FOR I=N-1 TO 1 STEP -1
120 P=P*I
130 NEXT I
140 PRINT:PRINT"FAKULTAET VON ";N;" IST ";P
150 PRINT:PRINT:PRINT"NOCH EINE RECHNUNG ? (J/N)";
:INPUT A$
160 IF A$="J" THEN CLS:GOTO 80
170 PRINT:PRINT:PRINT"ENDE":END
```

Variablenliste:

A$ = Stringvariable für Antworten
I = Laufindex (Faktoren des Produkts)
N = Zahl der Elemente
P = Zwischen- und Endprodukt

Programmbeschreibung:

Satz 1ø-7ø : Überschrift und Erläuterungen.

Satz 8ø-9ø : Anforderung der Inputinformation (Zahl der zu permutierenden Elemente N) und Belegung des Produktfeldes P mit dem Wert im Feld N als erstem Faktor der zu bildenden Produktkette.

Satz 1øø : Meldung,wenn N gleich oder größer 34 ist, weil dazu die Rechenkapazitäten nicht ausreichen und Sprung nach 15ø.

Satz 11ø-13ø : Bildung der Produktkette im Feld P.

Satz 14ø : Ergebnisausgabe.

Satz 15ø-17ø : Beendigung des Programms, es sei denn, es wird eine erneute Berechnung gewünscht (dann weiter bei Satz 8ø nach Löschen des Bildschirms).

9.13 Kombinationen

Problemstellung

Eine ähnliche Fragestellung wie im vorangegangenen Programm liegt vor, wenn berechnet werden soll, auf wieviele verschiedene Arten k Elemente aus n verschiedenen Elementen ausgewählt werden können.

Als Berechnungsformel ergibt sich:

$$\text{Anzahl} = \frac{n!}{k!(n-k)!}$$

Man nennt diese Größe Binomialkoeffizient und kürzt ihn ab mit folgender Schreibweise:

$$\binom{n}{k}$$

Er entspricht übrigens den Faktoren, die sich bei der Auflösung der sog. Binome ergeben:

$$(a+b)^2 = a^2 + 2ab + b^2$$

$$= 1a^2b^0 + 2a^1b^1 + 1a^0b^2$$

$$= \binom{2}{0}a^2b^0 + \binom{2}{1}a^1b^1 + \binom{2}{2}a^0b^2$$

oder:

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$= 1a^3b^0 + 3a^2b^1 + 3a^1b^2 + 1a^0b^3$$

$$= \binom{3}{0}a^3b^0 + \binom{3}{1}a^2b^1 + \binom{3}{2}a^1b^2 + \binom{3}{3}a^0b^3$$

Man sieht, daß jedes Binom sich in Summanden zerlegen läßt, wobei es immer ein Summand mehr ist als die Hochzahl des Binoms angibt (Hochzahl im letzten Beispiel war 3; Zahl der Summanden war 4).

Jeder Summand besteht aus 3 Faktoren:

- eine Konstante (der hier interessierende Binomialkoeffizient);
- Faktor "a hoch..." mit einer Hochzahl, die von der Binomhochzahl (hier 3) auf Ø sinkt;
- Faktor "b hoch ..." mit einer Hochzahl, die von Ø zur Binomhochzahl (hier 3) ansteigt.

Im hier interessierenden Binomialkoeffizienten wiederum steht als oberer Wert die Binomhochzahl (hier 3); als unterer eine ganze Zahl, die von links nach rechts von Ø bis zur Binomhochzahl ansteigt.

Das Programm, um das es hier geht, bestimmt beliebige Binomialkoeffizienten, so daß wir also mit diesem Programm leicht auch kompliziertere Binome bestimmen können (z.B. $(a+b)^{27}$).

Aber auch für Berechnungen kann dieses Programm genutzt werden, wie die folgenden beiden Programme zeigen.

Schließlich kann mit einem derartigen Programm z.B. folgende konkrete Frage beantwortet werden:

"8 Läufer starten zu einem 1ØØ-m-Lauf. Auf wieviele verschiedene Arten können die ersten drei Plätze besetzt werden?"

Antwort:

$$\binom{8}{3} = \frac{8!}{3!(8-3)!} = \frac{8!}{3!5!} =$$

$$= \frac{8 \cdot 7 \cdot 6}{3 \cdot 2 \cdot 1} = \underline{\underline{56}}$$

Programm:

```
10 REM Z12-KOMBINATIONEN
20 CLS
30 PRINT"PROGRAMM ZUR BESTIMMUNG DER ANZAHL DER"
40 PRINT"KOMBINATIONEN VON K AUS N ELEMENTEN."
50 PRINT:PRINT:PRINT"MAN SPRICHT IN DIESEM ZUSAMME
NHANG VON"
55 PRINT"    DEN SOG. BINOMIALKOEFFIZIENTEN."
60 PRINT:PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS,
1984"
70 PRINT:PRINT:PRINT:PRINT
80 INPUT"ZAHL DER ELEMENTE N         : ";N
90 INPUT"ZAHL DER AUSZUWAEHLENDEN K : ";K
100 Y=N-K
110 IF K=0 OR K=N THEN BK=1:GOTO 210
120 IF K>N/2 THEN H=Y:Y=K:K=H
130 GOSUB 500
140 PX=A:A=N
150 IF PX=1 THEN PN=N:GOTO 200
160 FOR I=N-1 TO N-K+1 STEP -1
170 A=A*I
180 NEXT I
190 PN=A
200 BK=PN/PX
210 PRINT:PRINT:PRINT"ANZAHL DER KOMBINATIONEN VON
 ";K;" AUS"
220 PRINT N;" ELEMENTEN IST    : ";BK
230 PRINT:PRINT:PRINT"ENDE DER BERECHNUNGEN":END
500 REM UP FAK
510 A=K
520 IF K=0 OR K=1 THEN A=1:GOTO 560
530 FOR I=K-1 TO 1 STEP -1
540 A=A*I
550 NEXT I
560 RETURN
```

Variablenliste:

A = Fakultät (Zwischen- und Endprodukt)
BK = Binimialkoeffizient
H = Hilfsgröße beim Tausch von K und Y
I = Laufindex (Faktor bei der Fakultätsberechnung)
K = Zahl der auszuwählenden Elemente
N = Zahl der Elemente
PN = Erste Fakultät im Binomialkoeffizient
PX = Zweite Fakultät im Binomialkoeffizient
Y = N-K

Programmbeschreibung:

Satz 1Ø- 7Ø : Überschrift und Erläuterungen

Satz 8Ø- 9Ø : Eingaben (Zahl der Elemente N und Zahl der Auszuwählenden K)

Satz 1ØØ : Belegung der Hilfsgröße Y

Satz 11Ø : Vereinfachte Berechnung für K=Ø oder K=N, weil definitionsgemäß gilt:

$$\binom{N}{Ø} = \binom{N}{N} = 1$$

Satz 12Ø : Vertauschen von K und (N-K) (=Y), falls K> N/2. Dies ist möglich, weil gilt:

$$\binom{N}{K} = \frac{N!}{K!(N-K)!} = \binom{N}{N-K}$$

Auf diese Weise werden die folgenden Berechnungen vereinfacht.

Satz 13Ø : Berechnung von K! im Unterprogramm 5ØØ

Satz 14Ø-19Ø : Berechnung der anderen Fakultät, die nach geeignetem Kürzen (siehe z.B. die Rennläufer-Aufgabe oben) noch übrig bleibt.

Satz 2ØØ-23Ø : Berechnung und Ausgabe des Binomialkoeffizienten und Beendigung des Hauptprogramms

---

Satz 5ØØ-56Ø : Berechnung von K! in einem Unterprogramm

9.14 Lotto

Problemstellung

Es soll ein Programm erstellt werden, welches die Chancen ausrechnet, beim Lottospiel (6aus 49) zu gewinnen.

Problemanalyse

Das Programm berechnet die Wahrscheinlichkeit für 3, 4, 5 oder 6 Richtige im Zahlenlotto (6 aus 49).

Die entsprechenden Wahrscheinlichkeiten berechnen sich gemäß der folgenden Formel:

$$P(X) = \frac{\binom{6}{X}\cdot\binom{43}{6-X}}{\binom{49}{6}} = \frac{B1*B2}{B3}$$

Dabei ist X die vom Benutzer anzugebende "Zahl der Richtigen".

Es muß hier also mit drei B-nomialkoeffizienten (B1, B2 und B3) gearbeitet werden, denen wir ja schon im vorangegangenen Kapitel begegnet waren.

Deshalb sind detailliertere Erläuterungen hier entbehrlich.

Programm

```
10 REM Z13-LOTTO
20 CLS
30 PRINT"PROGRAMM ZUR BESTIMMUNG DER WAHRSCHEIN-"
40 PRINT"LICHKEIT FUER 3,4,5 ODER 6 RICHTIGE IM"
50 PRINT TAB(11)"LOTTO (6 AUS 49)."
60 PRINT:PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS,
1984"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"DIESES PROGRAMM BERUECKSICHTIGT NICHT"
90 PRINT"DIE MOEGLICHKEIT 5 RICHTIGE MIT ZUSATZ-"
100 PRINT TAB(17)"ZAHL."
110 GOSUB 1000:REM WARTEN
120 INPUT "WIEVIELE RICHTIGE ";X
125 IF X=6 THEN B1=1:B2=1:GOTO 300
130 REM B1
135 IF X=5 THEN B1=6:B2=43:GOTO 300
140 Y=6-X
150 P=X:FOR I=X-1 TO 1 STEP -1
160 P=P*I:NEXT I:P1=P
170 P=Y:FOR I=Y-1 TO 1 STEP -1
180 P=P*I:NEXT I
190 P2=P
200 B1=720/(P1*P2)
210 REM B2
220 X=Y
230 P=43
240 FOR I=P-1 TO 43-X+1 STEP -1
250 P=P*I:NEXT I
260 P1=P
265 P=X
270 FOR I=X-1 TO 1 STEP -1
280 P=P*I:NEXT I
290 P2=P:B2=P1/P2
```

```
300 REM B3
310 P1=49:P2=6
320 FOR I=1 TO 5
330 P1=P1*(49-I)
340 P2=P2*(6-I)
350 NEXT I
360 B3=P1/P2:W=(B1*B2)/B3
370 CLS
380 PRINT"     ERGEBNIS :":PRINT:PRINT
390 PRINT"DIE WAHRSCHEINLICHKEIT FUER ";X
400 PRINT"RICHTIGE IM LOTTO BETRAEGT : "
410 PRINT:PRINT:PRINT TAB(5)W
420 PRINT:PRINT:PRINT:PRINT"NOCH EINE BERECHNUNG ?
(J/N)";:INPUT A$
430 IF A$="J" THEN CLS:GOTO 120
440 PRINT:PRINT:PRINT"ENDE DER BERECHNUNGEN":END
1000 REM UP WARTEN
1010 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !
"
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

Variablenliste :

A$ = Stringvariable für Antworteingaben
B1 = Erster Term in der Berechnungsformel
B2 = Zweiter Term in der Berechnungsformel
B3 = Dritter Term in der Berechnungsformel
P = Produkt in den Fakultätenberechnungen
P1 = Erste Fakultät im Binomialkoeffizienten
P2 = Zweite Fakultät im Binomialkoeffizienten
W = Wahrscheinlichkeit
X = Anzahl der "Richtigen"
Y = 6-X

Programmbeschreibung:

Satz 1Ø-1ØØ : Überschrift und Erläuterungen

Satz 11Ø : Sprung ins Unterprogramm 1ØØØ zum Warten

Satz 12Ø : Eingabe der Zahl der Richtigen

Satz 125 : Für X=6 gilt, daß B1=B2=1 (siehe Programm Z1Ø); weiter bei Satz 3ØØ

Satz 13Ø-2ØØ : Bestimmung von B1 gemäß der Überlegungen wie wir sie aus Programm Z1Ø schon kennen

Satz 21Ø-29Ø : Entsprechende Bestimmung des zweiten Binomialkoeffizienten B2

Satz 3ØØ-36Ø : Entsprechend B3

Satz 36Ø : Berechnung der Wahrscheinlichkeit W

Satz 37Ø-41Ø : Ausgabe des Ergebnisses

Satz 42Ø-44Ø : Beendigung des Hauptprogramms, es sei denn, es wird eine Alternativberechnung gewünscht

---

Satz 1ØØØ-1Ø4Ø : Unterprogramm zum Abwarten

9.15 Binomialverteilung

Problemstellung

Die Binomialverteilung ist eine Wahrscheinlichkeitsverteilung. Wahrscheinlichkeitsverteilungen wiederum dienen dazu, auf möglichst einfache Weise auch für kompliziertere Fragestellungen Wahrscheinlichkeiten zu berechnen.

Die binomische Verteilung ist auch deshalb besonders wichtig, weil sie als "Grundbaustein" für eine ganze Reihe darauf aufbauender Wahrscheinlichkeitsberechnungsverfahren dient.

Die binomische Verteilung antwortet auf die folgende Frage :

Wie groß ist die Wahrscheinlichkeit, daß bei n voneinander unabhängigen Versuchen x_i-mal das günstige Ereignis eintritt, wenn bei jedem Versuch nur zwei Ergebnisse möglich sind, und wenn die Eintrittswahrscheinlichkeit des günstigen Ereignisses bei einem Versuch P % beträgt ?

Ein einfaches Beispiel mag diese Frage etwas veranschaulichen :

Wie groß ist die Wahrscheinlichkeit, bei 1ø Münzwürfen z.B. 5 Mal "Kopf" zu werfen ?

Dies ist ein Anwendungsfall der binomischen Verteilung, weil bei jedem Versuch zwei Ereignisse möglich sind ("Kopf" und Zahl") und weil die Versuche (die Würfe) voneinander unabhängig sind.

Eine entsprechende Frage wäre beispielsweise :

Wie groß ist die Wahrscheinlichkeit, daß bei 5 Geburten genau 2 Mädchen zur Welt kommen ?

Problemanalyse

Zur Analyse des gestellten Problems beschäftigen wir uns einmal mit dem zuletzt erwähnten Beispiel :

Wenn bei fünf Geburten zwei Mädchen zur Welt kommen sollen, so wäre beispielsweise folgende "Kinder-Kombination" günstig im Sinne der Fragestellung :

M M K K K (M = Mädchen, K = Knabe)

Günstig wäre aber auch :

K K K M M

und viele andere Kombinationen.

Die Anzahl der insgesamt "günstigen" Kombinationen kann mit Hilfe der Permutationsberechnung gefunden werden, über die schon gesprochen wurde.

Ohne dies hier beweisen zu wollen, stellen wir fest, daß die Anzahl der Kombinationsmöglichkeiten von 2 Mädchen und 3 Knaben sich ergibt zu :

$$\text{Anzahl} = \frac{5!}{2! * 3!} = \frac{5*4*3*2*1}{2*1 * 3*2*1}$$

$$= \frac{5*4}{2*1} = 1\text{ø}$$

Allgemein gilt, daß die Anzahl der verschiedenen Anordnungsmöglichkeiten von n Elementen, von denen x_i Stück gleich sind (im Sinne von "günstig") und weitere $n-x_i$ ihrerseits gleich sind (im Sinne von "ungünstig") sich ergibt zu :

$$\text{Anzahl} = \frac{n!}{x_i!*(n-x_i)!}$$

Wir gehen bei der weiteren Problemanalyse davon aus, daß die Wahrscheinlichkeit einer Mädchengeburt P betrage (P = ø.5) - dies ist die Eintrittswahrscheinlichkeit des günstigen Ereignisses bei einem Versuch.

Betrachten wir nun noch einmal das Ergebnis :

M M K K K

Wie groß ist die Wahrscheinlichkeit eines solchen Ergebnisses bei fünf Versuchen ?

Betrachten wir zunächst die erste Mädchengeburt : Ihre Wahrscheinlichkeit ist offenbar gleich ø.5.

Auch die Wahrscheinlichkeit der zweiten Mädchengeburt ist ø.5.

Die Wahrscheinlichkeit nun dafür, daß die ersten beiden Geburten zusammen beides Mädchengeburten sind, beträgt dann ø.5 * ø.5 (allgemein P * P).

Die Wahrscheinlichkeit für eine Knabengeburt beträgt 1 - ø.5 (allgemein 1-P). Deshalb ergibt sich nach der gleichen Überlegung wie oben, daß die Wahrscheinlichkeit dafür, bei der dritten und bei der vierten und bei der fünften Geburt Knaben zu erhalten, sich ergibt zu :

$$(1 - \emptyset.5) * (1 - \emptyset.5) * (1 - \emptyset.5)$$

oder allgemein :

$$(1 - P) * (1 - P) * (1 - P)$$

Für die Kombination M M K K K erhalten wir deshalb insgesamt die folgende Wahrscheinlichkeit in allgemeiner Schreibweise :

$$P * P * (1 - P) * (1 - P) * (1 - P)$$

$$= P^2 * (1 - P)^3$$

$$= P^{x_i} * (1 - P)^{n-x_i}$$

In unserem Beispiel würde sich deshalb ergeben :

Wahrscheinlichkeit für M M K K K =

$$\emptyset.5^2 * (1 - \emptyset.5)^{5-2} = \emptyset.5^5 = 1/32$$

Nun sind aber in diesem Beispiel 1ø verschiedene M,K-Kombinationen günstig im Sinne der Fragestellung.

Deshalb ergibt sich die endgültige Wahrscheinlichkeit, weil für jede der 1ø Kombinationen sich die Einzelwahrscheinlichkeit 1/32 ergibt, zu :

Wahrscheinlichkeit für 2 Mädchen bei fünf Geburten = 1ø * 1/32 = 1ø/32 = 31.25 %

Aus diesem Beispiel können wir nun die allgemeine Berechnungsformel herleiten :

Die Wahrscheinlichkeit, bei n Versuchen x_i-mal das günstige Ereignis zu haben, wenn dessen Einzeleintrittswahrscheinlichkeit P beträgt, ergibt sich zu :

$$W = \frac{n!}{x_i!*(n-x_i)!} * P^{x_i} * (1 - P)^{n-x_i}$$

Diese Formel sieht recht kompliziert aus, läßt sich aber vergleichsweise leicht programmieren.

Programm:

```
10 REM Z14-BINOMIAL
20 CLS
30 PRINT"PROGRAMM ZUR BERECHNUNG VON WAHRSCHEIN-"
40 PRINT"LICHKEITEN MIT DER BINOMISCHEN VERTEI-"
50 PRINT TAB(17)"LUNG."
60 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT
80 PRINT"MIT DIESEM PROGRAMM BZW. MIT DER BINO-"
90 PRINT"MISCHEN VERTEILUNG KANN DIE FOLGENDE"
100 PRINT"FRAGE BEANTWORTET WERDEN :":PRINT
110 PRINT"WIE GROSS IST DIE WAHRSCHEINLICHKEIT,"
120 PRINT"DASS BEI N VERSUCHEN X-MAL DAS GUENSTI-"
130 PRINT"GE EREIGNIS EINTRITT, WENN BEI EINEM"
140 PRINT"VERSUCH DESSEN WAHRSCHEINLICHKEIT P BE-"
150 PRINT"TRAEGT ?"
160 GOSUB 1000:REM WARTEN
170 PRINT"DIESES PROGRAMM BENOETIGT ALS INPUT-"
180 PRINT"INFORMATIONEN : ":PRINT
190 PRINT:INPUT"ZAHL DER VERSUCHE              : ";
N
200 PRINT:INPUT"ZAHL DER GUENSTIGEN EREIGNISSE: ";
X
220 PRINT:PRINT"EINTRITTSWAHRSCHEINLICHKEIT BEI"
230 INPUT"EINEM VERSUCH      (DEZ.)      : ";P
240 REM BINOMIALKOEFFIZIENT
245 Y=N-X:Q=1-P
250 IF X=0 OR X=N THEN BK=1:GOTO 360
260 IF X>N/2 THEN H=Y:Y=X:X=H:G=Q:Q=P:P=G
270 F=X:GOSUB 500:REM FAK
290 PX=A:A=N
295 IF PX=1 THEN PN=N:GOTO 350
300 FOR I=N-1 TO N-X+1 STEP -1
310 A=A*I
320 NEXT I
```

```
330 PN=A
350 BK=PN/PX
360 W=BK*P^X*Q^Y
365 CLS
370 PRINT"      ERGEBNIS :":PRINT:PRINT
380 PRINT"DIE WAHRSCHEINLICHKEIT, DASS BEI ";N
390 PRINT"VERSUCHEN ";X;" MAL DAS GUENSTIGE"
400 PRINT"EREIGNIS EINTRITT, BETRAEGT : "
410 PRINT:PRINT:PRINT TAB(5)W
420 PRINT:PRINT:PRINT"NOCH EINE BERECHNUNG ? (J/N)
";:INPUT A$
430 IF A$="J" THEN CLS:GOTO 190
440 PRINT:PRINT:PRINT"ENDE DER BERECHNUNGEN":END
500 REM UP FAK
510 A=X
520 IF X=0 OR X=1 THEN A=1:GOTO 560
530 FOR I=X-1 TO 1 STEP-1
540 A=A*I
550 NEXT I
560 RETURN
1000 REM UP WARTEN
1010 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !
"
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
```

Variablenliste:

A = Produkt bei der Fakultätenberechnung
A$ = Stringvariable für Antworteingabe
BK = Binomialkoeffizient
F = Hilfsgröße (F=X)
G = Hilfsgröße beim Tausch von P und Q
H = Hilfsgröße beim Tausch von X und Y
I = Laufindex
N = Zahl der Versuche
P = Eintrittswahrscheinlichkeit des günstigen Ereignisses bei einem Versuch
PN = Erster Term des Binomialkoeffizienten
PX = Zweiter Term des Binomialkoeffizienten
Q = 1-P
W = Wahrscheinlichkeit
X = Zahl der günstigen Ereignisse
Y = N-X

Programmbeschreibung:

Satz 1ø-15ø : Überschrift und Erläuterungen.

Satz 16ø-23ø : Warten und danach Eingabe der Input-Informationen (Zahl der Versuche, Zahl der günstigen Ereignisse, Eintrittswahrscheinlichkeit).

Satz 24ø-35ø : Bestimmung des Binomialkoeffizienten :

245 : Hilfsfelder;
25ø : Ist X=ø oder X=1, dann ist der BK definiert als 1; weiter bei 36ø;
26ø : Ist X> N/2, so ist, um die Kürzungsmöglichkeiten auszunutzen, X mit N-X und P mit 1-P zu tauschen.
27ø : Bestimmung von x!;
29ø : Belegung des Nenners von BK (= PX) und Neubelegung von A, falls Var. A erneut benutzt wird(3øø);
295 : Ist der Nenner des BK = 1, so ist der Zähler des BK definiert als N; weiter bei Satz 35ø.
3øø-32ø : Bestimmung des Zählers des Binomialkoeffizienten BK;
35ø : Berechnung von BK.

Satz 36ø : Berechnung der gesuchten Wahrscheinlichkeit.

Satz 365-41ø : Ausgabe des Berechnungsergebnisses.

Satz 42ø-499 : Beendigung des Programms, es sei denn es wird eine erneute Berechnung gewünscht (dann nach Löschen des Bildschirms zurück zu Satz 19ø).

Satz 5øø-56ø : Unterprogramm zur Berechnung einer Fakultät.

Satz 1øøø-1ø4ø : Unterprogramm zum Warten.

9.16 Zentrales Grenzwert-Theorem (ZGWT)

Problemstellung

Gesetzt den Fall, man zieht aus einer vorliegenden Grundgesamtheit eine Zufallsstichprobe vom Umfang 5Ø (beispielsweise werden 5Ø zufällig ausgewählte Schüler einer Schule nach ihrer Zufriedenheit mit den Lehrern befragt).

Man kann dann in einer solchen Stichprobe unterschiedliche statistische Maßzahlen berechnen, wie beispielsweise ein arithmetisches Mittel (Durchschnittsalter der Schüler oder dergl.).

Wenn man nun alle verschiedenen Zufallsstichproben vom Umfang n=5ø ziehen würde, die überhaupt möglich sind (kein Mensch wird dies in der Praxis allerdings tun), dann erhält man ggf. sehr viele dieser Durchschnittswerte. Zeichnet man die Häufigkeitsverteilung all dieser Mittelwerte auf, so erhält man eine Verteilung, die sehr präzise dem Modell der Normalverteilung entspricht, unabhängig davon, wie die interessierende Variable (z.B. das Alter der Schüler) in der Grundgesamtheit verteilt ist, aus der die Stichproben gezogen wurden.

Auf diese Weise ist es möglich, wahrscheinlichkeitsbehaftete Aussagen über den Durchschnittswert in nur einer einzigen Zufallsstichprobe (und das ist die Praxis) zu machen.

Wir können dann errechnen, in welchen Grenzen der unbekannte Durchschnitt der Grundgesamtheit mit angebbarer Wahrscheinlichkeit (die unter Umständen recht hoch werden kann; z.B. 9ø, 95 oder 99 %) liegen muß, wenn ein Stichprobenmittelwert in bestimmter Größe festgestellt worden ist.

Diese Wahrscheinlichkeitsberechnungen sind nur deshalb möglich, weil wir die Gültigkeit der Normalverteilung unterstellen können, wie es das Zentrale Grenzwerttheorem von LaPlace und Liapunoff bewiesen hat :

Parameter von Zufallsstichproben folgen approximativ der Gauß'schen Normalverteilung, unabhängig davon, wie die jeweilige Verteilung in der Grundgesamtheit aussieht.

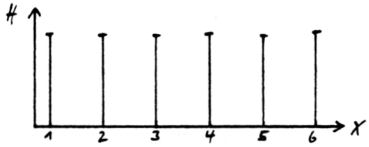
An einem einfachen Beispiel erweist das folgende Programm die Gültigkeit dieses Theorems. Es zeigt, daß Mittelwerte aus Zufallsstichproben approximativ normal verteilt sind, obwohl die Grundgesamtheit, aus der diese Stichproben gezogen werden alles andere als "normal" (im Sinne von Gauß) ist.

Zur Darstellung des Grenzwerttheorems lassen wir den Rechner 1øø Zufallsstichproben, jeweils vom Umfang n = 5ø, ziehen. Die Untersuchungsvariable seien Würfelwürfe.

Das heißt also,wir lassen den Rechner 5ø Mal würfeln und berechnen die durchschnittliche Augenzahl in dieser Stichprobe; danach würfelt er erneut 5ø Mal und berechnet den Durchschnitt, würfelt erneut 5ø Mal und berechnet den Durchschnitt usw. usw., und das alles insgesamt 1øø Mal.

Nach dieser Prozedur liegen 1øø Durchschnitte vor, die ihrerseits in Form einer Häufigkeitsverteilung durch den Rechner dargestellt werden können.

Es wird sich bei der Ausgabe dieser Häufigkeitsverteilung zeigen, daß sie keineswegs mehr dem Modell einer Würfel-Augenzahlverteilung entspricht, das ja schematisch folgendermaßen aussehen müßte :



Im Gegenteil : Es wird sich eine Verteilung ergeben, die der Normalverteilung recht nahe kommt (übrigens ist die Annäherung an das Modell der Gauß'schen Normalverteilung gemäß des Zentralen Grenzwerttheorems um so besser, je größer die Zufallsstichproben sind).

Zusätzlich zur graphischen Veranschaulichung der entstehenden Häufigkeitsverteilung, berechnen wir auch die kennzeichnenden Parameter, nämlich das arithmetische Mittel und die Standardabweichung. Dabei gilt :

$$\text{Arithmetisches Mittel} = \frac{\sum x_i}{n}$$

(entsprechend kann auch ein Mittelwert aller Stichprobenmittelwerte definiert werden);

$$\text{Standardabweichung} = \sqrt{\frac{1}{n}\sum (x_i - \bar{x})^2}$$

(dabei ist $\bar{x}$ das jeweilige arithmetische Mittel); entsprechend kann auch eine Standardabweichung aller Stichprobenmittelwerte definiert werden).

Im einzelnen ist also folgendes zu tun :

1. In dem zu entwickelnden Programm sollen 1øø Mal je 5ø Würfelwürfe simuliert werden und aus jeder 5øer Serie ist das arithmetische Mittel zu berechnnen.

2. Zusätzlich wird der Gesamtmittelwert berechnet und darüber hinaus als Streuungsmaß für alle insgesamt 5øøø Würfelwürfe die Standardabweichung.

3. In einem dritten Schritt erstellen wir dann die Häufigkeitsverteilung der 1øø erzeugten Stichprobenmittelwerte, indem die Mittelwerte in ein Klassifizierungsraster einsortiert werden, wobei die Klassen jeweils ø.1 Einheiten breit sind (dies wird sich als zweckmäßige Klassenbreite erweisen).

4. Im Anschluß daran wird der Mittelwert und die Standardabweichung dieser Häufigkeitsverteilung bestimmt und diese beiden Maße werden mit den entsprechenden Werten, wie sie in Punkt 2. berechnet wurden, verglichen.

5. Schließlich wird die Häufigkeitsverteilung der Stichprobenmittelwerte auch der Anschaulichkeit halber graphisch ausgegeben.

Programm:

```
10 REM Z15-ZGWT
20 CLS
30 PRINT"PROGRAMM ZUR ILLUSTRATION DES ZENTRALEN"
40 PRINT"GRENZWERTTHEOREMS (NORMALVERTEILUNG VON"
50 PRINT TAB(15)"GAUSS)."
60 PRINT:PRINT:PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT:PRINT:PRINT
80 PRINT"DIESES PROGRAMM FUEHRT 100 SERIEN ZU JE"
90 PRINT"50 WUERFELWUERFEN DURCH UND BESTIMMT"
100 PRINT"DEN MITTELWERT IN JEDER SERIE."
110 PRINT:PRINT"UEBER DIESE MITTELWERTE WIRD DANN
EINE"
120 PRINT"HAEUFIGKEITSVERTEILUNG ERSTELLT."
122 GOSUB 1100:REM WARTEN
125 N=100
130 DIM AM(N),KA(11),KE(11),H(11),W(N,50)
140 FOR I= 1 TO N
150 S=0
160 FOR J=1 TO 50
170 W(I,J)=INT(RND(1)*6+1)
180 S=S+W(I,J):SS=SS+W(I,J)
190 NEXT J
200 AM(I)=S/50
210 NEXT I
```

```
215 MW=SS/(N*50):PRINT:PRINT"GESAMTMITTELWERT  = "
;MW:PRINT:PRINT
216 GOSUB 1100:REM WARTEN
217 FOR I=1 TO N:FOR J= 1 TO 50
219 S1=S1+(W(I,J)-MW)^2:NEXT J:NEXT I
223 S9=SQR(S1/(N*50))
225 PRINT:PRINT"STANDARDABWEICHUNG = ";S9
226 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
227 REM HAEUFIGKEITSVERTEILUNG
230 GOSUB 1000: REM ABWARTEN
240 PRINT"MITTELWERT IN 50-ER    ANZAHL"
250 PRINT"       SERIEN"
260 FOR I=1 TO 30:PRINT"-";:NEXT I:PRINT
270 FOR K=1 TO 10
280 KA(K)=3+(K-1)*0.1
290 KE(K)=KA(K)+0.1
300 NEXT K
310 FOR I=1 TO N
320 FOR K=1 TO 10
330 IF AM(I)<KE(K) THEN H(K)=H(K)+1:GOTO 350
340 NEXT K
345 H(11)=H(11)+1
350 NEXT I
360 FOR K=1 TO 10
370 PRINT KA(K);TAB(6)" BIS UNTER ";KE(K);TAB(25)H
(K)
380 NEXT K
```

```
385 PRINT TAB(11)"UEBER ";KE(10);TAB(25)H(11)
390 REM MITTELWERT DER SPV
400 S=0
410 FOR J=1 TO N
420 S=S+AM(J)
430 NEXT J
440 A=S/N
450 PRINT:PRINT:PRINT"MITTELWERT DER VERTEILUNG :
";A
455 REM STREUUNG DER SPV
460 S=0
470 FOR J=1 TO N
480 S=S+(AM(J)-A)^2
490 NEXT J
500 SS=SQR(S/N)
510 PRINT:PRINT:PRINT"STREUUNG DER VERTEILUNG   :
";:PRINT USING "###.##";SS
515 GOSUB 1000:REM ABWARTEN
520 PRINT:PRINT:PRINT"     GEGENUEBERSTELLUNG : ":
PRINT:PRINT
530 PRINT TAB(20)"WUERFEL   GAUSS":PRINT
560 PRINT"MITTELWERT";:PRINT TAB(20) USING "##.##"
;MW;:PRINT TAB(30) USING "##.##";A
570 PRINT:PRINT"STREUUNG";:PRINT TAB(20) USING "##
.##";S9;:PRINT TAB(30) USING "##.##";SS
580 GOSUB 1000:REM WARTEN
```

```
600 REM GRAPHIK
610 PLOT 0,80
620 DRAW 600,80
630 FOR K=1 TO 11
640 E=20:A=E-H(K):IF A<3 THEN A=3
645 LOCATE 3*K-1,A-2:PRINT H(K)
650 FOR J= A TO E
660 LOCATE 3*K,J:PRINT CHR$(143)
670 NEXT J
680 NEXT K
730 LOCATE 5,23
740 PRINT"ENDE DER AUSGABE":END
1000 REM UP ABWARTEN
1010 LOCATE 5,23:PRINT"BITTE EINE TASTE DRUECKEN !
1020 A$=INKEY$:IF A$="" THEN 1020
1030 CLS
1040 RETURN
1100 REM UP WARTEN
1110 PRINT:PRINT:PRINT TAB(7)"BITTE WARTEN, ICH RE
CHNE":PRINT:PRINT
1120 RETURN
```

Variablenliste:

A = Mittelwert aller Mittelwerte,
später Anfangsadresse zum Zeichnen

AM = Stichprobenmittelwert

A$ = Stringvariable (Antworteingabe)

E = Endadresse beim Zeichnen

H = Häufigkeit

I = Laufindex (Serie Nr.)

J = Laufindex (Wurf Nr.)

K = Laufindex (Klasse)

KA = Klassenanfangspunkt

KE = Klassenendpunkt

MW = Gesamtmittelwert

N = Zahl der Würfe

S = Seriensumme

SS = Gesamtsumme (Später Standardabweichung)

S1 = Quadratsumme

S9 = Standardabweichung

W = Würfelwurf

Programmbeschreibung:

Satz 1ø-12ø : Überschrift und Erläuterungen.

Satz 122 : Warten (per Unterprogramm 11øø).

Satz 125-13ø : Angabe der Zahl der zu ziehenden Stichproben und Dimensionierung der benötigten Variablenfelder (es wird dabei ein Stichprobenumfang von 5ø unterstellt).

Satz 14ø-21ø : Erzeugung von 1øø Stichproben zu je 5ø Würfelwürfen :

15ø : Belegung eines Summenfeldes S mit dem Wert ø;
16ø-19ø : Es wird 5ø Mal gewürfelt.
17ø : Würfelwurf;
18ø : Bildung der stichprobeninternen Augensumme S und der Gesamtsumme SS;
2øø : Bestimmung des Stichprobenmittelwertes AM.

Satz 215 : Errechnung des Gesamtmittelwertes MW und Ausgabe.

Satz 216 : Abwarten.

Satz 217-225 : Bildung der Gesamt-Standardabweichung als Maß der Streuung der Augenzahlen und Ausgabe dieses Maßes.

219 : Addition aller quadrierten Abweichungen der Augenzahlen vom Gesamtmittelwert im Feld S1;
223 : Berechnung der Standardabweichung;
225 : Ausgabe des Ergebnisses.

Satz 227-23ø : Warten.

Satz 24ø-26ø : Ausgabe einer Tabellenüberschrift für die Häufigkeitsverteilung der Stichprobenmittelwerte.

Satz 27ø-3øø : Bestimmung der Klassenuntergrenzen (KA) und der Klassenobergrenzen (KE) für die zu erzeugende Häufigkeitsverteilung (Klassenbreite = ø.1).

Satz 31ø-35ø : Einordnen der Stichprobenmittelwerte in die Häufigkeitsklassen;

31ø : Vorgabe eines Mittelwerts (von den insgesamt 1øø Mittelwerten);
32ø : Vorgabe einer Häufigkeitsklasse (von den insgesamt 1ø Klassen);
33ø : Ist der Mittelwert kleiner als die Klassenobergrenze KE, so ist er in diese Klasse einzuordnen (H=H+1); dann weiter bei 35ø (nächster Mittelwert); wenn nicht, weiter bei 34ø (nächste Klasse);
34ø : nächste Klasse;
345 : Besetzung einer Restklasse

35ø : nächster Mittelwert

Satz 36ø-385 : Ausgabe der Häufigkeitsverteilung

Satz 39ø-45ø : Berechnung des Mittelwerts aller Stichprobenmittelwerte (= Mittelwert der sog. Stichprobenverteilung) :

4øø : Summenfeld S = ø;
41ø-43ø : Aufsummierung aller Stichprobenmittelwerte AM;
44ø : Berechnung des Gesamtmittels A;
45ø : Ausgabe von A.

Satz 455-51ø : Berechnung der Streuung aller Stichprobenmittelwerte (= sog. Stichprobenfehler) :

46ø : Summenfeld S = ø;
47ø-49ø : Bildung der Summe aller quadrierten Abweichungen zwischen Stichprobenmittelwerten und Gesamtmittelwert;
5øø : Berechnung der Standardabweichung (= Streuung der Stichprobenverteilung);
51ø : Ausgabe.

Satz 51ø-57ø : Gegenüberstellung der einzelnen errechneten Maßzahlen.

Satz 58ø : Warten.

Satz 6øø-7øø : Erzeugung des graphischen Abbilds der Häufigkeitsverteilung :

61ø-62ø : Zeichnen eines waagrechten Strichs;
63ø : Vorgabe einer Häufigkeitsklasse;

64ø : Bestimmung des Endpunktes
und des Anfangspunktes des
zu zeichnenden Balkens, der
die jeweilige Häufigkeit re-
präsentieren soll;

645 : Ausgabe der Häufigkeit;

65ø-67ø: Zeichnen des Balkens

68ø : Nächste Klasse

Satz 73ø-74ø : Beendigung des Programms

Satz 1øøø-1ø4ø : Unterprogramm Warten

Satz 11øø-112ø : Unterprogramm Wartemeldung

Wenn dieses Programm gestartet wird, dann benötigt es zunächst ca. 3 Minuten, um das erste Zufallsergebnis zu produzieren :

GESAMTMITTELWERT = 3.5218

Nach weiteren 5 Minuten meldet der Rechner dann :

STANDARDABWEICHUNG = 1.7ø573292

Bei einem völlig korrekten Würfel liegt übrigens der zu erwartende Mittelwert bei 3.5 und die zu erwartende Standardabweichung bei 1.7ø8.

Danach ergibt sich die folgende Häufigkeitsverteilung, die wohlgemerkt ein Zufallsergebnis darstellt :

MITTELWERT IN 5ø-ER SERIEN			ANZAHL
3	BIS UNTER	3.1	5
3.1	BIS UNTER	3.2	6
3.2	BIS UNTER	3.3	6
3.3	BIS UNTER	3.4	8
3.4	BIS UNTER	3.5	15
3.5	BIS UNTER	3.6	19
3.6	BIS UNTER	3.7	17
3.7	BIS UNTER	3.8	1ø
3.8	BIS UNTER	3.9	14
3.9	BIS UNTER	4	ø
	UEBER	4	ø

MITTELWERT DER VERTEILUNG : 3.52
STREUUNG DER VERTEILUNG : .23

Weiterhin gibt der Rechner aus :

GEGENÜBERSTELLUNG

	WUERFEL	GAUSS
MITTELWERT	3.52	3.52
STREUUNG	1.71	ø.23

Wir erkennen, daß die Streuung der Stichprobenverteilung, die - wie schon erwähnt wurde - annäherungsweise der Gauss'schen Normalverteilung entspricht, sehr viel kleiner ist als die der ursprünglichen Verteilung. Darauf beruht letzten Endes die Präzision, die bei Hochrechnungen, ausgehend von Zufallsstichproben, erreicht werden kann.

Das Zentrale Grenzwerttheorem beweist u.a. die folgenden Beziehungen :

Mittelwert der Stichprobenverteilung		Mittelwert der Grundgesamtheit
3.52	=	3.52

Streuung der Stichprobenverteilung		Streuung der Grundgesamtheit dividiert durch die Wurzel aus dem Stichprobenumfang
ø.23	~	$1.71/\sqrt{5ø}$
	=	1.71/7.ø7...
	=	ø.242

Schließlich produziert das Programm noch eine graphische Darstellung der oben vorgeführten Häufigkeitsverteilung für Stichprobenmittelwerte, wobei die einzelnen Häufigkeiten als Balken unterschiedlicher Länge dargestellt werden.

9.17 Wahlen

Problemstellung

Das Programm veranschaulicht graphisch in Form eines Balkendiagramms Wahlergebnisse.
Dazu muß der Benutzer die prozentualen Stimmenanteile für CDU, SPD, F.D.P., Grüne und Sonstige eingeben.

Problemanalyse

Eine Problemanalyse ist bei dieser recht einfachen Aufgabe wieder entbehrlich.

Programm:

```
10 REM Z16-WAHLEN
20 CLS
30 PRINT"PROGRAMM ZUR GRAPHISCHEN DARSTELLUNG"
40 PRINT TAB(9)"VON WAHLERGEBNISSEN."
50 PRINT:PRINT:PRINT:PRINT
60 PRINT TAB(9)"PROF.DR.W.VOSS, 1984"
70 PRINT:PRINT:PRINT
80 PRINT"EINZUGEBEN SIND DIE WAHLERGEBNISSE IN %"
90 PRINT:PRINT:PRINT
95 DIM F(5),G(5)
100 INPUT "CDU/CSU      : ";F(1)
110 INPUT "SPD          : ";F(2)
120 INPUT "F.D.P.       : ";F(3)
130 INPUT "GRUENE       : ";F(4)
140 INPUT "SONSTIGE     : ";F(5)
150 CLS
160 PLOT 0,110:DRAW 600,110
180 FM=0:FOR I=1 TO 5:IF F(I)>FM THEN FM=F(I)
190 NEXT I
200 FOR I=1 TO 5:G(I)=(17/FM)*F(I)
210 G(I)=INT(G(I)+0.5):NEXT I
```

```
230 GOSUB 2000:REM ZEICHNEN
250 LOCATE 1,22
260 PRINT"CDU   SPD   FDP   GRU   SONST"
270 FOR J=1 TO 5:PRINT TAB(6*(J-1))F(J);:NEXT J
280 PRINT:PRINT"ENDE":END
2000 REM UP ZEICHNEN
2010 FOR J=1 TO 5
2020 B=17:A=18-G(J)
2030 FOR I= A TO B
2040 LOCATE (J-1)*6+2,I:PRINT CHR$(143)
2050 LOCATE (J-1)*6+3,I:PRINT CHR$(143)
2060 NEXT I
2070 NEXT J
2080 RETURN
```

Variablenliste:

A = Startzeile der Zeichnung

B = Endzeile der Zeichnung

F = Prozentanteile

FM = Maximalwert

G = Transformierte Prozentanteile

I = Laufindex

J = Laufindex

Programmbeschreibung:

Satz 1ø- 9ø : Überschrift und Erläuterungen

Satz 95 : Dimensionierungen

Satz 1øø-14ø : Eingabe der Inputinformationen

Satz 15ø-16ø : Löschen des Bildschirms und Zeichnen eines waagrechten Strichs

Satz 18ø : Aufsuchen der maximalen Prozentangabe

Satz 2øø-21ø : Umrechnen der Prozentangaben derart, daß der Maximalwert den auf dem Bildschirm zur Verfügung stehenden Platz ausnutzt

Satz 23ø : Sprung ins UP 2øøø zum Zeichnen

Satz 25ø-27ø : Ausgabe der Texte CDU, SPD, F.D.P., GR, SO und der Prozentangaben unterhalb der Zeichnung

Satz 28ø : Ende des Hauptprogramms

Satz 2øøø-2ø9ø : Unterprogramm zum Zeichnen der Balken

Satz 2ø1ø-2ø2ø:
Bestimmung von Anfang- und Endadresse des Balkens

Satz 2ø3ø-2ø7ø:
Zeichnen von inversen Blanks an die per LOCATE bestimmten Stellen

Satz 2ø9ø: Rücksprung

Ausblick

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.

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.

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.

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.

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 :

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 Programmlauf abgebrochen wird, dann hilft immer noch ein Blick in das Benutzerhandbuch.

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.

A

Abbildung, optische 163
Abfrage 28
ABS 25
Addition 23
Anfangswert 32
Arbeitsschritt 41
Archimedes 151
Argument 24
Array 11
Ausgabe 12
Ausprägung 1Ø

B

BASIC 7, 17
BASIC-Anweisung 95, 179
BASIC-Graphikanweisung 136
Betriebssystem 12
Bevölkerungsentwicklung 245
Bildschirm 13
Bildschirmzone 21
Binomialverteilung 347
Biologie 211
Bit 9
Blockgraphik 158
Bruttobetrag 8Ø
Buchstabe 9
Bundesländer 282
Byte 9

C

CAT 35
Chemie 95
CHR$ 25, 97
CLS 36
CONT 36
COS 25, 3Ø9
Cosinus 3Ø9
CTRL 398

D

DATA 99
Datei 11
Daten 8, 12
Dateneingabe 99
Datensatz 11
Datenverarbeitung 8
DIM 1øø
Dimensionierung 1øø
Diskette 13, 35
Division 23
Dreieck, rechtwinkliges 42
Dreisatz 81
Dreisatzrechnung 8ø
Drucker 13

E

E 74
Eingabe 12
Eingabefehler 4ø
Element, chemisches 122
Ellipse 316
END 18
Endlosschleife 36
Endwert 32
Englisch-Vokabeln 196
Englisch-Vokabeltest 196
ENTER-Taste 24
Erdkunde 231
Ergebnis 41
Ergebnisausgabe 17
ESC 36, 98
Eva-Prinzip 42, 152
EXP 25
Exponentialdarstellung 92

F

Farbadresse 141
Farbe 97, 98, 99
Federwaage 144
Feld 1ø
File 11
Flußdiagramm 14, 41
FOR...TO 32
Französisch-Vokabeln 189
Fremdsprache 179
Funktion 24, 25
Funktionsname 24
Funktionsweise 12

G

Gauß 32ø
Gerade 296
Geschichte 231
Gesetz, Ohm'sches 173
GGT 5ø
Gitter 293
Gleichung, quadratische 68
GOSUB 18ø
GOTO 3ø
Graphik 132, 293
Graphik, hochauflösende 132
Graphik-Programmierung 132
Graphikzeichen 96
Grenzwerttheorem 358

H

Häufigkeitsverteilung 273
Hauptstädte 238
Hypothek 262
Hypothekentilgung 262

I

IF...THEN 28
Indizierung, doppelte 1ø1
Informationen 8
Informationseingabe 26
Inhaltsverzeichnis 35
INKEY$ 18ø
INPUT 26, 98
INT 25

J

Jahreszahlen 232
Jahreszahlen, historische 232

K

Kassette 13
KGV 5ø
Kombinationen 336
Kommando 12, 2ø
Korrektur 24
Kräfteparallelogramm 168
Kreis 299

L

Laden 34
Länder 238, 282
Laufvariable 32
Leerzeile 18
LET 23, 98
Linse 163
LIST 2ø
LOAD 34
LOG 25, 313
Logarithmus 313
Loop 32
Lotto 342

M

Mathematik 39, 291
Menü 174
Menütechnik 174
Mittel, arithmetisches 269
Mittelwert 27, 269
Multiplikation 23

N

Name 1ø
Nation 245
Nettobetrag 8ø
NEW 21
NEXT 32
Normalgraphik 158
Normalverteilung 32o, 358

O

Ökologie 211
OPEN 35

P

Pendel 157
Pendelbewegung 157
Periodensystem 122
Permutationen 332
Physik 131
Potenzierung 23
Primzahl 59
Primzahlenprüfung 59
PRINT 18
PRINT USING 173
Problemanalyse 14, 41
Problemvorstellung 41
Programm 41
Programmanweisung 11, 12
Programmbeschreibung 41
Programmieren 16
Programmschleife 32
Programmverzweigung 28
Prozentrechnung 8ø
Punkt 297, 298
Pythagoras 41

Q

Quadratzahl 3ø6

R

READ 99
Reaktion, chemische 1ø9
Reaktionsgleichung 1ø9
Rechengenauigkeit 92
Rechenoperation 23
Rechnen, stöchiometrisches 116
Rechnung 18
Regression 324
REM 37
RESTORE 1ø2
RESTORE-Taste 98, 298
RETURN 181
RND 25, 85
RUN 19

S

Satz 17
Satznummer 28
SAVE 35
Schleife 32
Schrittweite 32
SHIFT 98
SIN 25, 398
Sinus 3Ø3, 3Ø9
Sinuslinie 3Ø3
Sonderzeichen 9
Sortieren 2Ø3
Sortierprogramm 2Ø7
Spaltenindex 1Ø1
Speicher, externer 13, 34
Speichern 34
Speicherplatz 1ØØ
Speicherstelle 1Ø
Sprache 179
Sprung 28
Sprung, bedingter 28
Sprung, unbedingter 28
SQR 25
Statement 17
STEP 33
Steuerbetrag 8Ø
Stöchiometrie 116
STOP 36
Strich 136
String 11, 18
Stringvariable 22
Subtraktion 23
Symbol 9

T

TAB 21
TAN 25, 3Ø9
Tangens 3Ø9
Tastatur 13, 96
Trend 325

U

Umweltverschmutzung 223

V

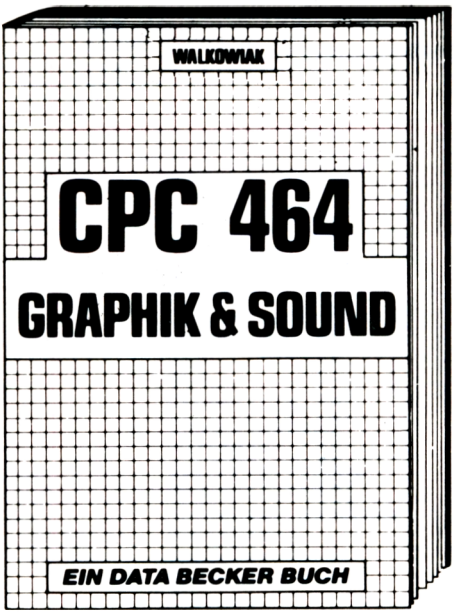
Variable 1ø, 22
Variable, ganzzahlige 22
Variable, indizierte 1øø
Variable, reelle 22
Variablenliste 41
Variablenname 18, 22
VC 1541 35
Verarbeitung 12
Verben 182
Verben, englische 182
Verben, unregelmäßige 182
Verzweigung 28
Vokabeln 189
Vokabeltest 196

W

Wachstum 212
Wachstum, gebremstes 217
Wachstum, ungebremstes 212
Wahlen 376
Wassermolekül 1ø3
Wert 1ø
Wertzuweisung 22
Wirtschaft 255
Wirtschaftskraft 282
Würfel 85
Wurzel 3ø6

Z

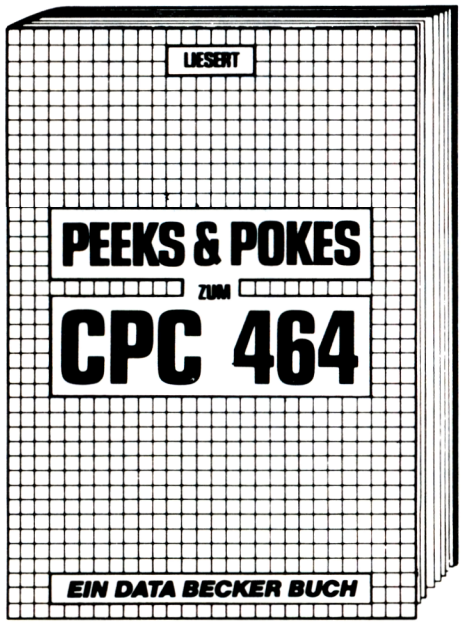
Zahl 18
Zahl, Euler'sche 74
Zeilenindex 1ø1
Zentraleinheit 13
ZGWT 358
Ziffer 9
Zins 256
Zinsrechnung 256
Zufallszahl 85



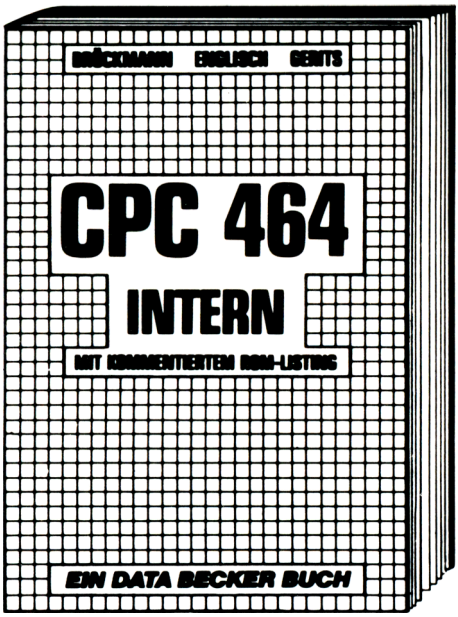
In diesem erstklassigen Buch wird gezeigt, wie man die außergewöhnlichen Grafik- und Soundmöglichkeiten des CPC 464 nutzt. Natürlich mit vielen interessanten Beispielen und nützlichen Hilfsprogrammen. Aus dem Inhalt: Grundlagen der Grafikprogrammierung, Sprites, Shapes und Strings, mehrfarbige Darstellungen, Koordinatentransformation, Verschiebungen, Drehungen, Rotation, 3-D-Funktionsplotter, CAD, Synthesizer, Miniorgel, Hüllkurven und vieles mehr.

Walkowiak
CPC 464 Graphik & Sound
220 Seiten, DM 39,-
ISBN 3-89011-050-9

Wer die wichtigen Peeks und Pokes zum CPC 464 kennen und anwenden will, der findet hier umfassende Information. Sie reicht vom Adreßbereich des Prozessors über Betriebssystem unter Interpreter bis hin zur Einführung in die Maschinensprache. Dazu präzise Programmierhilfen, sinnvolle Routinen sowie reichlich Material zu den Themen Grafikfunktionen, Massenspeicherung und Peripherie, Tricks und Formeln in BASIC, RAM-PAGES.



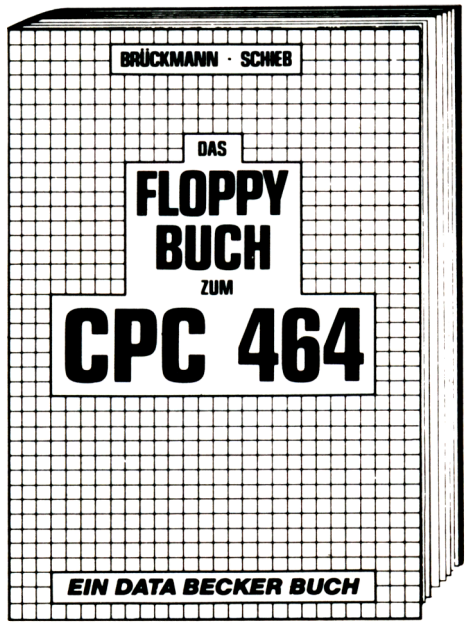
Liesert/Schieb
Peeks & Pooks zum CPC 464
ca. 220 Seiten, DM 29,-
erscheint Mai 1985
ISBN 3-89011-092-4



Wirklich alle Geheimnisse des CPC 464 lüftet dieses Standardwerk, das für den Fortgeschrittenen BASIC-Programmierer unentbehrlich, für den Assembler-Programmierer ein absolutes Muß ist. Neben dem ausführlich dokumentierten und kommentierten BASIC-ROM-Listing enthält es umfangreiche Kapitel zu Speicheraufteilung, Prozessor, Besonderheiten des Z 80, Gate Array, Video-Controller und Video-Ram, Soundchip, Schnittstellen, Betriebssystem, Routinennutzung, Character-Generation, BASIC-Interpreter und mehr.

Brückmann/Englisch/Gerrits
CPC 464 intern mit kommentiertem ROM-Listing
548 Seiten, DM 69,-
ISBN 3-89011-080-0

Alles über Diskettenprogrammierung vom Einsteiger bis zum Profi. Natürlich mit ausführlichem ROM-Listing (Betriebssystem), einer äußerst komfortablen Dateiverwaltung, einem hilfreichen Disk-Monitor und einem ausgesprochen nützlichen Disk-Manager. Dazu eine Fundgrube verschiedener Programme und Hilfsroutinen, die das Buch für jeden Floppy-Anwender zur Pflichtlektüre machen.



Brückmann/Schieb
Das Floppy-Buch zum CPC 464
ca. 250 Seiten, DM 49,-
erscheint Mai 1985
ISBN 3-89011-093-2

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 am CPC 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.

ISBN 3-89011-040-1