Ian Stewart

Das Tor zur Computer-Welt

Commodore 64 für Fortgeschrittene

Hobby-Paperbacks

mvg verlag

Ian Stewart

Das Tor zur Computer-Welt
Commodore 64 für Fortgeschrittene

Ebenfalls in der Reihe
„Das Tor zur Computerwelt“:

Commodore 64 für Anfänger
(ISBN 3-478-02240-1)

Diese Reihe wird fortgesetzt.

Ian Stewart

Das Tor zur Computer-Welt

Commodore 64 für Fortgeschrittene

Deutsche Bearbeitung:
Dirk und Stefan Petry

mvg verlag

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Stewart, Ian:
Das Tor zur Computer-Welt / Ian Stewart. [Aus d. Engl. übertr. von Dirk u. Stefan Petry]. -
Landsberg am Lech: Moderne Verlagsgesellschaft (mvg-Hobby-Paperbacks)
Commodore 64 für Fortgeschrittene. - 1985.
Orig.-Ausg. u.d.T.: Stewart, Ian: Gateway to computing with the Commodore 64
ISBN 3-478-02250-9

Titel der Originalausgabe: Gateway to Computing with the Commodore 64 - Book 2

First published in Great Britain 1984 by Shiva Publishing Limited.

Aus dem Englischen übertragen von Dirk und Stefan Petry.

8910 Landsberg am Lech
Umschlaggestaltung: Hendrik van Gemert
Titelbild: Shiva Publishing Ltd., Cheshire
Satz: Fotosatz Roßkopf GmbH & Co. KG, 8901 Königsbrunn
Druck und Bindearbeiten: Presse-Druck, Augsburg
Printed in Germany 020 250/385602
ISBN 3-478-02250-9

Inhaltsverzeichnis

Einleitung

Die Reihe „Das Tor zur Computer-Welt" soll junge Leute (und solche, die jung geblieben sind) auf unterhaltende und verständliche Art in die Grundlagen des Programmierens einführen. Die Bände dieser Reihe sind für die besonders verbreiteten Heimcomputer erhältlich, so z. B. für den Commodore 64 und den ZX-Spectrum. Buch 1 behandelt im wesentlichen die Bedeutungen der wichtigsten BASIC-Befehle. Obwohl der vorliegende Band auch einige neue Befehle vorstellt, werden doch Programmiertechnik und Programmstruktur betont.

Manchmal habe ich Angst davor, eine von Maschinen gesteuerte Welt zu bauen, die kaum noch ein Mensch versteht. Die Fähigkeit, Programme zu schreiben, ist zwar nur ein Aspekt des Computerwissens, aber ein sehr wichtiger: Sie hilft uns, das Geheimnisvolle des Computers zu durchschauen. Und dabei können wir auf keine Hilfsmittel verzichten.

Diese Reihe basiert auf der Überzeugung, daß es möglich ist, sich mit einem Thema ernsthaft zu beschäftigen, ohne dabei ernst zu sein. Lernen kann Spaß machen. Schwierige Inhalte können leichter verstanden werden, wenn man sie fröhlich und abwechslungsreich präsentiert. Hier eine kleine Kostprobe der Gestalten, die Du auf den folgenden Seiten antreffen wirst: Squire Stoatthrostle, Sherlock Holmes und Dr. Watson, Bernard und Ermintrude Klöbner . . . - um nur einige zu nennen.

Mit ihrer Hilfe (gelegentlich auch Widerstand) gewinnt der fortgeschrittene Leser dieses Buches Schritt für Schritt Einblick in zahlreiche programmiertechnische Themen, wie zum Beispiel:

- Datenspeicherung
- Computer-Logik
- Fehlersuche und -beseitigung
- Strings - wie man damit umgeht
- Schleifenschachtelung
- Datenmanipulation.

„Das Tor zur Computer-Welt - Commodore 64 für Fortgeschrittene" geht auf diese Programmierprobleme ein; zugleich gibt es dem Leser Rätsel auf, stellt ihm Aufgaben und übt mit ihm die neu erlernten Fertigkeiten, hilft ihm, sie anzuwenden. Die Antworten werden jeweils am Ende der Kapitel gegeben.

Unter den vielen Beispielprogrammen sind:

- Wetter-Daten-Analyse
- Primzahlengenerator
- Toms Telefonbuch
- Pickaxos Rechteckzeichner.

Das erste Kapitel faßt kurz die Hauptthemen aus Band 1 zusammen; der Rest des Inhalts wurde sorgfältig ausgewählt, so daß dieses Buch für den, der schon die ersten BASIC-Klippen umfahren hat, auf einem angemessenen Niveau steht.

Die Reihe „Das Tor zur Computer-Welt" zeichnet sich - das hoffe ich sehr - durch folgende Eigenschaften aus:

- Sie ist verständlich,
- und sie macht vor allen Dingen Spaß!

1

Holmes erinnert sich . . .

Während ich die lange gewundene Treppe zum Privatkrankenzimmer des Königlichen Hals-, Nasen- und Ohren-Hospitals von Bethnal Green erklomm, kam mir ein einzelner Satz nicht mehr aus dem Sinn: Moriarty, dieser Teufel, möge er in der Hölle schmoren! Es war nämlich Professor Moriarty, der Napoleon des Verbrechens, gewesen, der meinen berühmten Kollegen Sherlock Holmes in diese schreckliche Lage gebracht hatte. Ich fragte mich, inwieweit ein ungeschützter menschlicher Schädel einem zehn Kilogramm schweren Jade-Frosch standhalten konnte. Ich muß gestehen, ich war auf das Schlimmste gefaßt.

Ich war also sehr erleichtert, als mir die Krankenschwester erlaubte, in die schwach beleuchtete Kammer einzutreten, und ich Holmes sah, der aufrecht im Bett saß und hastig seine Pfeife unter dem Kissen versteckte. Aber sein erster Satz jagte mir einen Schauer über den Rücken.

„Wer zum Teufel sind Sie?" sagte er.

„Holmes! Sie kennen mich nicht mehr?"

„Nein", seufzte er, „ich kenne mich ja selbst nicht mehr! Ich habe nicht die leiseste Ahnung, wer ich bin."

„Amnesie!" rief ich.

„Hocherfreut, Sie kennenzulernen, Mr. Amnesia", sagte Holmes plötzlich. „An dem Schnitt des Knopflochs an Ihrem Rockaufschlag erkenne ich, daß Sie Ihre Anzüge bei Popodopoulos, einem Schneider in Athen, kaufen. Sie haben einen typisch griechischen Haarschnitt, und Ihre Augenbrauen hängen etwas unausgeglichen herunter, was man oft bei Menschen sieht, die einen Großteil ihres Lebens mit langen Seereisen verbringen.

Sind Sie vielleicht Mr. Stavros Z. Amnesia, der Großreeder?"

„Nein, nein", unterbrach ich ihn, obwohl ich froh war zu sehen, daß er nichts von seiner scharfen Beobachtungsgabe und seinen bemerkenswerten Kombinationstalenten eingebüßt hatte. „Ich heiße John H. Watson, Dr. med., und Sie sind der berühmte Sherlock Holmes. Um auf die Amnesie zurückzukommen, . . ."

„Ich kann mich an überhaupt nichts mehr erinnern", sagte Holmes.

„Richtig, alter Junge!" Es war ein gutes Zeichen, daß Holmes nicht seinen medizinischen Wortschatz vergessen hatte.

„Wofür bin ich berühmt?" fragte Holmes.

„Sie sind der bekannteste und der fähigste Detektiv der Welt", erklärte ich ihm. „Und außerdem sind Sie einer der sorgfältigsten Debugger der Computerszene." (Meine Leser werden sich ohne Zweifel an Sherlock Holmes' entscheidende Beiträge zu jenem vorzüglichen Band mit Namen „Das Ohr am Computerfeld für Anfänger" oder so ähnlich erinnern.)

„Was ist ein Computer, Dr. Holmes?" fragte er leise.

„Nein, Sie sind Holmes. Ein Computer, mein Lieber, ist ein Gerät, das Informationen verarbeitet." Er sah mich verdutzt an. Ich setzte also von neuem an: „Eine Maschine, die Daten manipuliert. Ein künstliches Gehirn. Ein Zahlen-Schlukker." Nichts zu machen. „Ein Durchbruch in technischer und erzieherischer Hinsicht: Er sollte in keinem Haus fehlen." Keine Reaktion. „Die größte Erfindung seit der Sicherheitsnadel." Immer noch nichts. „Eine Kunststoffschachtel mit vielen Knöpfen, die man zum Spielen von Telespielen gebraucht."

Plötzlich grinste er. „Ach, einer von denen", rief er.

Endlich ein Zeichen, daß sein Gedächtnis langsam zurückkehrte! „Aber Sie, Holmes", fuhr ich fort, „Sie haben mehr getan als nur Spiele gespielt: Sie haben PROGRAMME geschrieben."

Sein Grinsen verbreiterte sich. „Ja, ja, natürlich! Wie faszinierend, Dr. Whatsit." Das Grinsen verschwand. „Ich habe nur ein Problem . . ."

Ich wartete.

„Ich kann mich nicht im geringsten daran erinnern, was ein Programm ist."

Programme

„Ein Programm, Holmes, ist eine Liste von Befehlen, die der Computer ausführen soll, um eine bestimmte Aufgabe zu erfüllen. Es wird in einer der vielen speziell für diesen Zweck vorgesehenen *Sprachen* geschrieben. Eine sehr verbreitete und beliebte Computersprache ist **BASIC.** Das ist die Abkürzung für -"

„**B**lutige Anfänger sinken **i**n perfektes **C**haos."

„Nein Holmes: Es heißt **B**eginners' **A**ll-purpose **S**ymbolic **I**nstruction **C**ode, was soviel heißt wie Symbolische Allzweck-Programmiersprache für Anfänger."

„Fast", sagte Holmes trocken.

Ein leicht verständlicher Abschnitt, in dem der Autor seinen Lesern das ins Gedächtnis zurückruft, was sie aus Band 1 noch wissen sollten.

„Ich werde Ihnen einige Eigenschaften der Sprache BASIC ins Gedächtnis zurückrufen, Holmes", sagte ich in meiner hilfsbereiten Art. „Vielleicht bringt das Ihr Gedächtnis auf Trab." Ich durchsuchte meine Brieftasche, während ich weitersprach. „Befehle werden in BASIC mit Hilfe von bestimmten *Befehlswörtern* gegeben: solche sind zum Beispiel **RUN, LIST, STOP, NEW, FOR, NEXT, TO, IF, THEN, INPUT, PRINT, LET, GOTO.** Bringt das irgendetwas zum Klingeln?"

Er dachte angestrengt nach. „Sind diese Wörter vielleicht eine Art Verkehrszeichen für diese neumodischen Kutschen ohne Pferde?"

In diesem Moment hatte ich den Eindruck, irgendetwas zu riechen - es erinnerte mich an verbrannte Federn. Aber Holmes' Antwort hatte mich abgelenkt. „Nicht ganz, Holmes", sagte ich lächelnd. „Aber Sie werden wärmer."

„Verteufelt komisch", sagte er, „ich glaube, Sie haben recht."

„Eigentlich sind es BASIC-Befehlswörter, Holmes." Endlich hatte ich die Karte, die ich suchte, gefunden - sie hatte in meiner Hosentasche gesteckt. „Dies ist eine sogenannte BASIC-Referenz-Karte; sie wurde aus einem sehr interessanten Buch zusammengestellt, ich glaube, sein Name war ‚Wachposten am Computertor - Buch 74'." Ich zeigte Holmes die Karte.

Watsons handliche Referenz-Karte

RUN Programm ausführen.

LIST Die Befehle eines Programms auflisten.

STOP Programm unterbrechen.

NEW Altes Programm im Speicher löschen.

FOR/TO/NEXT Wird zum Aufbau von *Schleifen* benutzt. Die Schleife beginnt mit einem Befehl wie **FOR** N=1 **TO** 1Ø und endet mit **NEXT** N. Die Befehle zwischen den Zeilen werden in diesem Beispiel 10mal ausgeführt, wobei N nacheinander die Werte 1-1Ø annimmt.

IF/THEN Verzweigungsbefehl. **IF** (Aussage) **THEN** (Handlung) bewirkt, daß die Handlung dann ausgeführt wird, wenn die Aussage wahr ist. Ist sie falsch, fährt der Computer mit der nächsten Zeile fort.

INPUT Befähigt den Benutzer, dem Computer Zahlen oder Strings mitzuteilen. Ein String ist eine Reihe von Zeichen wie zum Beispiel "Q%7(" oder "FRED". Die Anführungszeichen sind nicht Teil des Strings; sie zeigen nur Anfang und Ende des Strings an.

PRINT Daten auf dem Bildschirm ausgeben.

LET Einer Variablen einen Wert zuweisen. Eine Variable ist ein mit einem Namen versehener Speicherbereich, der zum Speichern von Zahlen oder Strings verwendet wird. Der Wert (= Inhalt) einer Variablen kann verändert werden. **LET** K = 17 weist der numerischen Variablen K den Wert 17 zu. LET B$ = "FRED" weist der Stringvariablen B$ den Wert "FRED" zu. Beachte das $-Zeichen bei Stringvariablen.

GOTO Sprung zu einer anderen Programmzeile. In gehobenen Kreisen verpönt. Aber unter Freunden durchaus anwendbar.

Holmes' Interesse erwacht

„Nun, Holmes?" erkundigte ich mich vorsichtig. „Erinnern Sie sich an diese BASIC-Befehle und ihren Gebrauch?" Er mußte sich erinnern! Tat er es nicht, dann würden wir nicht weit über Kapitel 1 hinauskommen.

Holmes begann sich unbehaglich hin und her zu bewegen. „Ich glaube schon, Whatsit. Sie kommen mir irgendwie bekannt vor. Was sollte ich tun, wenn ich mich nicht an sie erinnern kann?"

„Ich schlage vor, Sie lesen einen einzigartigen Bestseller, den ich in meiner Wohnung 221B Baker Street liegen habe, ‚Computergeld', Band 492, oder so ähnlich." (Ich war schon immer stolz auf mein exzellentes Gedächtnis, und das nicht zu unrecht, wie Sie sehen.)

„Ja!" rief Holmes. „Ich glaube, mein Gedächtnis kehrt tatsächlich zurück! Ich sehe ein eigenartiges Bild ... es sieht aus wie ... wie ein russischer Spion, der fast völlig von einem großen Kaugummi-Ballon verdeckt wird.

„Das ist das Buch, Holmes!"

„Aha. Bleibt nur noch ein Loch in meinem Gedächtnis, Whatsit. Wie bin ich - warten Sie, Whatsit, brät hier jemand Fasan? Ich könnte schwören, daß - aber das ist nicht wichtig, ich bevorzuge ohnehin Gänsebraten. Was ich Sie fragen wollte: Welche Umstände haben mich in dieses Krankenhaus gebracht?"

„Moriarty, dieser Teufel, möge er in der Hölle schmoren!" Ich erklärte Holmes kurz, wie er sich zwischen den Rhododendren in Grating Towers, dem Stammsitz des Duke of Westhamptonshire, verborgen hatte, nachdem ihm durch einen Informanten mitgeteilt worden war, daß Professor Moriarty den verwegenen Diebstahl der unschätzbar wertvollen, mit Perlen besetzten Manschetten-

knöpfe des Dukes plane. Holmes war gerade dabei gewesen, den Dieb in flagranti zu fassen, als ein Komplize ihm mit einem zehn Kilogramm schweren Jade-Frosch auf den Kopf schlug. (Ich selbst habe ja den Butler im Verdacht, es könnte aber auch der Polizist gewesen sein!)

„Natürlich!" rief Holmes. „Und dann hat sich dieser Teufel Moriarty (möge er in der Hölle schmoren) ohne Zweifel zusammen mit seinem Komplizen und der Beute davongemacht."

„Richtig!"

„Reichen Sie mir meine Hosen, Whatsit! Wir müssen den Schurken sofort aufspüren!"

„Natürlich, Holmes. Solange seine Spur noch heiß ist!"

Holmes begann zu schwitzen. Fieber? Oder vielmehr der brennende Wunsch, diesen Teufel Moriarty (möge er in der Hölle schmoren) zu fangen?

„Whatsit?"

„Ja, Holmes?"

„Was, sagten Sie, sei heiß?"

„Die Spur, Holmes!"

„Oh, ich dachte ... Wissen Sie, ich fühle mich nämlich etwas seltsam. Es könnte sein, daß ich fiebere. Hier ist es sehr warm, meinen Sie nicht auch?"

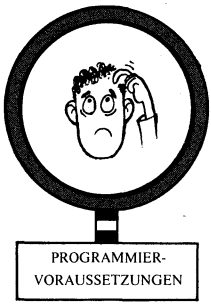
„Ich weiß nicht, Holmes. Draußen wütet ein Schneesturm, und die Oberin hat wie üblich die Fenster weit offen gelassen. Ich frage nur kurz, ob ..."

In diesem Moment brannte das Kissen lichterloh. Holmes sprang sehr behende vom Bett, wobei er eine Rauchspur, ausgehend vom Hinterteil seines Flanell-Pyjamas, hinter sich her zog und den betroffenen Körperteil mit beiden Händen festhielt. Es schien, daß er seine Pfeife so eilends vor den prüfenden Augen der Schwester verbor-

gen hatte, daß er vergessen hatte, sie zu löschen, bevor er sie unter das Kissen steckte.

Nun, als Mediziner muß ich sagen, daß es mich freute, einen solchen unumstößlichen Beweis für den Satz zu sehen: „Rauchen gefährdet Ihre Gesundheit."

Verschnaufpause



Die kleine Episode von Watson und Holmes hat den Sinn, Dich an die Hauptaussagen des ersten Bandes dieser Reihe zu erinnern. Du mußt dieses Buch nicht gelesen haben; aber Du solltest die BASIC-Befehlswörter aus Watsons handlicher Referenz-Karte kennen und mit ihnen umgehen können. Du solltest außerdem in der Lage sein, ein Programm einzugeben, es zu verbessern und es laufen zu lassen. Und auch einen Fehler mittels einfacher Such-Techniken solltest Du aufspüren können.

Es folgen nun einige Probleme, die Dein Können prüfen. Überlege nochmals genau, ob Du alle Antworten verstanden hast bevor Du im nächsten Kapitel weiterliest.

Rudis Addierer

Rudi Ruhsam wurde beauftragt, ein Programm zu schreiben, das die Zahlen von 1 bis 1000 addiert. Während er sich etwas hinter der Kaffeemaschine ausruht, kannst Du versuchen, ihm die Arbeit abzunehmen.

Raffkes Kreditkarte

Ralf Raffke, Geschäftsführer der Volksbankfiliale Kleinhinterwald, hat den Entschluß gefaßt, allen seinen Kunden eine Armenian-Excess-Kreditkarte (niemals ohne) zukommen zu lassen, vorausgesetzt, sie haben mindestens 500 DM auf ihrem Konto. Er hat Dich als Programmierer angeworben (für ein enormes Gehalt, denn er kennt keinen besseren als Dich), damit Du ein Programm schreibst, das Namen und Kontostand eines Kunden als Eingabe entgegennimmt und daraufhin dem Kassierer mitteilt, ob dem betreffenden Kunden eine Kreditkarte geschickt werden muß oder nicht. Du mußt zusätzlich das Programm als Schleife gestalten, so daß es immer bereit ist, die nächsten Eingaben anzunehmen; wird aber der Name „MUNGO" eingegeben, dann soll das Programm abbrechen. (Frag mich bitte nicht, warum gerade „MUNGO" o. k.?)

Wirst Du's schaffen, dieses enorm hohe Gehalt zu verdienen?

Wahlmaschine

Das folgende Programm wurde einem Software-Paket der Firma Softbirne für den Stork-37-Mikrocomputer entnommen; es ist für die Computerunterstützte Wahlabwicklung gedacht. Die Benutzer geben ihre Wahl ein, der Computer zählt die Stimmen und gibt aus, wer gewonnen hat. Wie die meisten Softbirne-Produkte hat auch dieses Programm einige Fehler.

Finde und behebe sie, indem Du zum Beispiel die Trockenlauf-Technik anwendest.

```
10 PRINT"COMPUWAHL AUTOMATIKDEMOKRATIE P
AKET V.7"
20 PRINT"NAME DES KANDIDATEN 1?"
30 INPUTN$
40 PRINT"PARTEI DES KANDIDATEN 1?"
50 INPUTP$
60 PRINT"NAME DES KANDIDATEN 2?"
70 INPUTN$
80 PRINT"PARTEI DES KANDIDATEN 2?"
90 INPUTP$
100 PRINT"GUTEN MORGEN, LIEBE WAEHLER!"
110 PRINT"DIES IST IHRE CHANCE, IHRE"
120 PRINT"DEMOKRATISCHEN GRUNDRECHTE AUS
ZUUEBEN."
130 PRINT"SIE HABEN DIE WAHL ZWISCHEN"
140 PRINT"1: ";N$;" VON DER PARTEI ";P$
150 PRINT"UND"
160 PRINT"2: ";N$;" VON DER PARTEI ";P$
170 PRINT"GEBEN SIE BITTE 1 ODER 2 EIN."
180 LETT1=0
190 LETT2=0
200 INPUTC
210 IFC=1THENLETT1=T1+1
220 IFC=2THENLETT2=T2+1
230 PRINT"WAR DAS DIE LETZTE STIMMABGABE
?"
240 INPUTA$
250 IFA$="NEIN"THENGOTO100
260 IFT1>T2THENPRINT"KANDIDAT 1 GEWINNT
FUER DIE PARTEI";P$
270 IFT1<T2THENPRINT"KANDIDAT 2 GEWINNT
FUER DIE PARTEI";P$
```

Antworten

Rudis Addierer

```
10 LETSUMME=0
20 FORN=1TO1000
30 LETSUMME=SUMME+N
40 NEXTN
50 PRINT"GESAMTSUMME =";SUMME
```

Raffkes Kreditkarte

```
10 PRINT"ARMENIAN EXCESS KREDITKARTEN-ZU
TEILUNG"
20 PRINT"KUNDENNAME?"
30 INPUTNAME$
40 IFNAME$="MUNGO"THENGOTO100
50 PRINT"AKTUELLER KONTOSTAND?"
60 INPUTKS
70 IFKS<500THENPRINT"AN DEN KUNDEN ";NAM
E$;" KEINE KARTE SCHICKEN!"
80 IFKS>=500THENPRINT"KARTE AN DEN KUNDE
N ";NAME$;" SCHICKEN!"
90 GOTO20
100 PRINT"DIESES PROGRAMM WIDMETE IHNEN
DIE"
110 PRINT"ARMENIAN EXCESS INCORPORATED"
120 PRINT"(NIEMALS OHNE)"
```

Wahlmaschine

Das Programm gliedert sich in drei Abschnitte: Die Zeilen 10-160 bereiten den Wahlvorgang vor; 170-250 bilden den Hauptteil, in dem die Stimmen entgegengenommen und gezählt werden; 260-270 schließlich entscheiden, welcher Kandi-

dat gewonnen hat. Das Programm kann Abschnitt für Abschnitt verbessert werden.

Der erste Schritt ist, das Programm laufen zu lassen, Testwerte einzugeben und die Ergebnisse zu untersuchen. Wir könnten zum Beispiel folgende Eingaben machen:

Name des Kandidaten 1: TILDE TIEFGANG
Partei des Kandidaten 1: VOLKSFRONT DER TONTAUBENZUECHTER
Name des Kandidaten 2: RUDOLF RUEPEL
Partei des Kandidaten 2: VEREINIGTE EGOISTEN

Und sofort geht etwas schief. Der Computer sagt uns, wir hätten die Wahl zwischen:

1: RUDOLF RUEPEL VON DER PARTEI VEREINIGTE EGOISTEN
UND
2: RUDOLF RUEPEL VON DER PARTEI VEREINIGTE EGOISTEN

Dies scheint die Kandidatin 1, Tilde Tiefgang, etwas zu benachteiligen. Wir lassen uns aber davon nicht beirren, geben eine Stimme für unsere Freundin Frau Tiefgang ein und teilen dann dem Computer mit, daß die Wahl beendet ist. Wir erhalten daraufhin die Meldung:

KANDIDAT 1 GEWINNT FUER DIE PARTEI VEREINIGTE EGOISTEN

Die Nummer des Kandidaten stimmt zwar, aber nicht die Partei.

Hmmmm.

Meine Verbesserungsversuche führten zu folgendem Ergebnis:

1. In den Zeilen 30, 50, 70 und 90 werden die Variablen N$ und P$ zweimal mit verschiede-

nen Bedeutungen verwendet. Ein Trockenlauf zeigt, daß diese Variablen am Ende immer Kandidat 2 und Partei 2 enthalten. Um dies zu beheben, müssen wir die Variablen trennen; dies geschieht zum Beispiel, indem wir M$ und P$ für Kandidat und Partei 1 vorsehen und N$ und Q$ für Kandidat und Partei 2. Das bedeutet:

in Zeile 30 N$ zu M$ ändern
in Zeile 90 P$ zu Q$ ändern

Konsequenterweise müssen im ersten Abschnitt dann noch einige weitere Änderungen vorgenommen werden:

in Zeile 140 N$ zu M$ ändern
in Zeile 160 P$ zu Q$ ändern

Der erste Teil des Programms funktioniert jetzt, aber insgesamt liegen noch grobe Fehler vor.

2. Ein Trockenlauf des zweiten Abschnittes zeigt, daß die Zeilen 180 und 190 die Variablen T1 und T2 bei jedem Schleifendurchlauf von neuem auf Null setzen. Diese Zeilen sollten VOR der Schleife liegen:

   Zeile 180 nach 92 verschieben
   Zeile 190 nach 94 verschieben

3. Wenn Du beim INPUT-Befehl in Zeile 250 "NEIN" als Antwort eingibst, erhältst Du eine Menge unerwünschter Ausgaben. Der Sprung führt zur falschen Zeile.
   In Zeile 250 **GOTO** 100 zu **GOTO** 170 ändern

4. Im mittleren Abschnitt scheint nun alles in Ordnung zu sein, aber im letzten sieht es ganz wüst aus. Wie auch immer gewählt wurde, das Ergebnis lautet immer, daß der Sieger von der

Partei 1 stammt. Deshalb müssen wir natürlich

in Zeile 270 P$ zu Q$ ändern

5. Es scheint nun alles einwandfrei zu arbeiten, das heißt aber nicht, daß wir das Programm vollständig „entwanzt" haben. Was passiert, wenn beide Kandidaten gleich viele Stimmen erhalten (gib zum Test jedem eine)? Es wird nichts ausgegeben. Wir brauchen daher eine zusätzliche Zeile:

```
280 IF T1 = T2 THEN PRINT "WAHL
    UNENTSCHIEDEN!"
```

2
Mehr Schleifen

Nachdem Du nun die einfache **FOR. . .NEXT**-Schleife kennst, können wir uns auch mit schöneren Schleifen beschäftigen. Ich meine:

(a) Schleifen, die nicht in Einerschritten weiterzählen, und
(b) Schachtelungen mehrerer Schleifen.

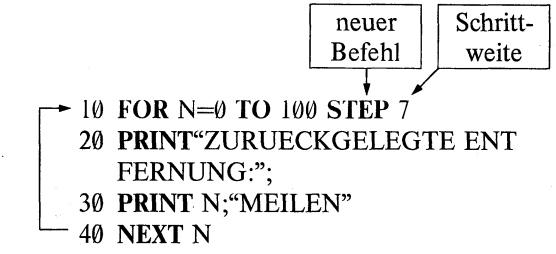
Beginnen wir mit (a).

Sieben-Meilen-Stiefel

Eine alte Geschichte erzählt von einem Mann, der ein Paar Sieben-Meilen-Stiefel besaß - Stiefel, die mit einem Schritt sieben Meilen zurücklegten. (Eine Meile sind ungefähr 2 Kilometer.) In BASIC gibt es einen Befehl:

STEP

(=Schritt), der in Verbindung mit der **FOR...NEXT**-Schleife die Laufvariable in beliebigen Schrittweiten auf- oder abwärts zählen kann. Das folgende Programm zum Beispiel druckt die Entfernungen aus, die mit Sieben-Meilen-Stiefeln zurückgelegt werden können.



Die Schleife funktioniert genauso wie bisher, nur daß N jetzt in Siebenerschritten aufwärts gezählt wird:

0, 7, 14, 21, . . ., 98

Bei 98 ist die Schleife beendet, da der nächste Wert, 105, größer ist als der Endwert der Schleife, 100. (Sieben-Meilen-Stiefel sind unpraktisch, wenn man nur die eine Meile zur nächsten Pizzeria gehen will.) Ein weiteres Beispiel:

FOR N=25 **TO** 35 **STEP** 2

ergibt die Zahlen:

25, 27, 29, 31, 33, 35

(Es tauchen nur die ungeraden Zahlen zwischen Anfangs- und Endwert auf, da der Anfangswert ungerade ist und in Zweierschritten weitergezählt wird.)

FOR N=1 **TO** 1000 **STEP** 100

ergibt:

1, 101, 201, 301, 401, 501, 601, 701, 801, 901

Mausschlaus Intelligenztest

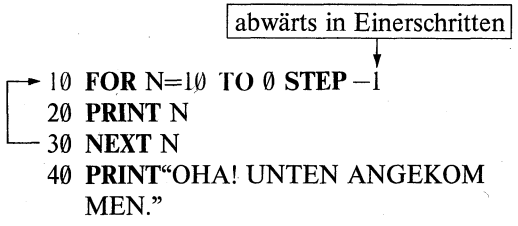
Martin Mausschlau, der mysteriöse Mathematiker, sucht nach **FOR**. . .**NEXT**-Schleifen, die diese Zahlenfolgen ausgeben:

(a) 2, 4, 6, 8, 10, 12, 14, 16
(b) 10, 13, 16, 19, 22, 25, 28, 31, 34
(c) 50, 60, 70, 80, 90, 100
(d) 514, 537, 560, 583, 606, 629

Kannst Du ihm helfen?

Countdown

Du kannst auch negative Schrittweiten angeben, um abwärts zu zählen:



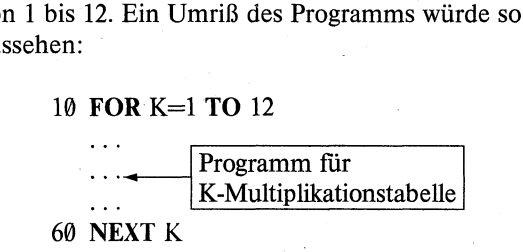
Geschachtelte Schleifen

Stell Dir vor, Du willst das Einmaleins von 1*1 bis 12*12 ausdrucken. Na los! Stelle es Dir vor. Tu's für mich.

Du könntest zwölf verschiedene Programme schreiben und beginnen mit:

```
10 FOR N=1 TO 12
20 PRINT N;"*";1;"=";N*1
30 NEXT N
```

Jetzt müßtest Du nacheinander die 1 in Zeile 20 zu 2, 3, ..., 12 ändern. Das wäre ein ziemlicher Aufwand. Und dabei sehen alle Programme *fast* gleich aus. Hier kann eine zweite Schleife nützliche Dienste erweisen; ihre Laufvariable K zählt von 1 bis 12. Ein Umriß des Programms würde so aussehen:



Jetzt ist es sehr leicht, das obige Programm zur Ausgabe des Einmaleins von K zu verallgemeinern:

```
20 FOR N=1 TO 12
30 PRINT N;"*";K;"=";N*K
40 NEXT N
```

Beachte die Strichpunkte!

Wenn wir das in den Umriß einpassen und ein wenig aufräumen, erhalten wir:

äußere Schleife

innere Schleife

```
10 FOR K=1 TO 12
20 FOR N=1 TO 12
30 PRINT N;"*";K;"=";N*K
40 NEXT N
50 PRINT
60 NEXT K
```

fügt eine Leerzeile ein

Ja, das ist faszinierend! Wir haben das Problem mit zwei ineinander geschachtelten Schleifen gelöst.

Vielleicht willst Du erkunden, was passiert, wenn man die beiden NEXTs in die falsche Reihenfolge bringt; nämlich so

```
10 FOR K=1 TO 12
20 FOR N=1 TO 12
30 PRINT N;"*";K;"=";N*K
40 NEXT K
50 PRINT
60 NEXT N
```

Du wirst sehen, daß das nicht funktioniert.

Wenn eine Schleife in eine andere eingelagert wird:

(a) Vergewissere Dich, daß sie auch vollständig, mit den **NEXT**s in der richtigen Reihenfolge, eingelagert ist.

(b) Verwende verschiedene Laufvariablen.

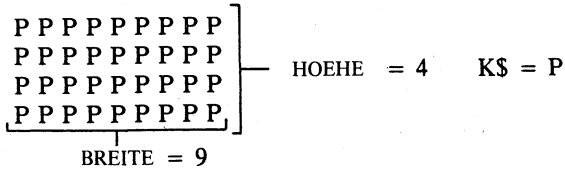
Aufgabe

Laut Martin Mausschlau, dem magischen Mathematiker, gibt es genau eine zweistellige Zahl, die zweimal das Produkt ihrer beiden Stellen ist. Schreibe ein Programm, das diese Zahl ermittelt. (Zweistellige Zahlen sind die Zahlen von 10 bis 99. Ein Produkt ist das Ergebnis einer Multiplikation. Versuchen wir es mit 72. Zweimal das Produkt der Stellen ist 7*2, das ist zweimal 14 oder 28. Das ist NICHT gleich 72, also ist 72 nicht die gesuchte Zahl. Jetzt mußt Du nur noch 89 weitere Zahlen untersuchen . . .)

Ein Tip: Verwende zwei geschachtelte Schleifen, für jede Stelle eine.

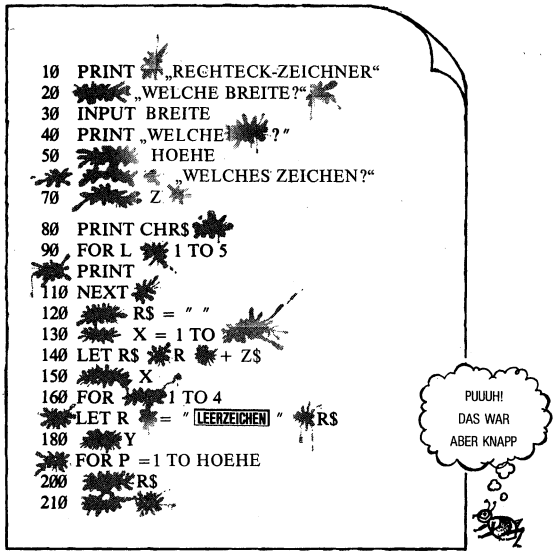
Pickaxos Programm

Pablo Pickaxo, der progressive Maler, hat ein Programm geschrieben, das Rechtecke ausgibt, die durch Wiederholen eines einzelnen Buchstabens erzeugt werden.



Er verwendet drei Variablen: HOEHE und BREITE, um die Größe des Rechtecks zu bestimmen, und K$, um den Buchstaben auszuwählen.

Leider ist eine Schar Termiten über sein Programm gezogen. Pablo konnte das Programm zwar erfolgreich entwanzen (vielmehr: enttermiten), dabei wurde es aber teilweise unleserlich. Sein Hammer wurde auch etwas schmutzig. Kannst Du die fehlenden Teile ausfüllen?



Antworten

Mausschlaus Intelligenztest

(a) **FOR** N=2 **TO** 16 STEP 2
(b) **FOR** N=10 **TO** 34 STEP 3
(c) **FOR** N=50 **TO** 100 STEP 10
(d) **FOR** N=514 **TO** 629 STEP 23

Aufgabe

```
10 FOR A=1 TO 9
20 FOR B=0 TO 9
30 LET X=10*A+B
40 IF X=2*A*B THEN PRINT"ICH HABE SIE: "
;X
50 NEXT B
60 NEXT A
```

Pickaxos Programm

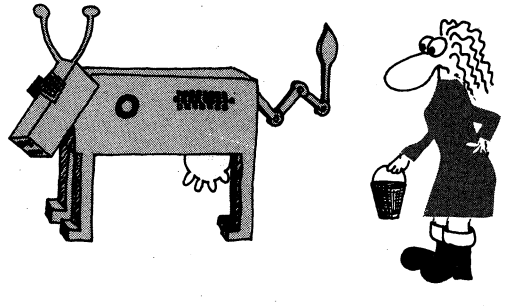
```
10 PRINT"RECHTECK-ZEICHNER"
20 PRINT"WELCHER BREITE?"
30 INPUT BREITE
40 PRINT"WELCHE HOEHE?"
50 INPUT HOEHE
60 PRINT"WELCHES ZEICHEN?"
70 INPUT Z$
80 PRINT CHR$(147)
90 FOR L=1 TO 5
100 PRINT
110 NEXT L
120 LET R$=""
130 FOR X=1 TO BREITE
140 LET R$=R$+Z$
150 NEXT X
160 FOR Y=1 TO 4
170 LEYT R$=" "+R$
180 NEXT Y
190 FOR P=1 TO HOEHE
200 PRINT R$
210 NEXT P
```

3
READ und DATA

Millie McHeck, die manische Milchmagd, bekannt für die Einführung des Millie-Liters, wurde von ihrem Vater Harald McHeck gebeten, seinen Bauernhof auf Computerverwaltung umzustellen. Millie beschließt, zunächst einmal eine Liste aller Kühe anzufertigen. Jede Kuh hat eine Seriennummer, die auf ihrem Ohr vermerkt ist. Die vollständige Liste wäre für unser Beispiel zu lang, hier ist ein Ausschnitt:

JOSEPHINE	22443
XANTIPPE	69888
KLARA	71450
BERTA	22222
ROSALINDE	93665

Das Problem ist: Wie lege ich diese Informationen in einem Programm ab . . . und wie bekomme ich sie wieder, wenn ich sie brauche?



Millie hat ein Datenspeicher-Problem. Es wird mit diesen BASIC-Befehlen gelöst:

DATA READ RESTORE

Mit ihrer Hilfe kann man Datenlisten in ein Programm integrieren und einzelne Informationen wiederfinden.

Verdatete Kühe

DATA-Einträge sind Zahlen oder Strings, ganz wie Du willst. Einzelne Einträge einer Datenliste müssen durch Kommata getrennt werden. Strings müssen nicht unbedingt mit Anführungszeichen umgeben werden. Millies Problem wird - zumindest teilweise - durch diese Programmzeilen gelöst:

```
10 DATA JOSEPHINE,22443
20 DATA XANTIPPE,69888
30 DATA KLARA,71450
40 DATA BERTA,22222
50 DATA ROSALINDE,93665
```

usw.

Wenn Millie diese Liste verwenden will, muß sie auf einzelne Einträge zugreifen können. Das geschieht mit dem Befehl

READ

(=Lies), der in dieser Form verwendet wird:

READ Variable

Er weist der angegebenen Variablen den Wert des „nächsten" Eintrags der **DATA**-Liste zu.

Was meine ich mit „nächsten"? Nun, es ist so:

Stell Dir vor, die Liste hätte eine *Markierung,* die vom Computer angebracht wurde. Zu Beginn eines Programmlaufs zeigt diese Markierung auf den Anfang der Liste:

10 **DATA** JOSEPHINE,22443
...

Nach jedem **READ**-Befehl setzt der Computer die Markierung auf den folgenden Eintrag der Liste. Nach dem ersten **READ** sieht die Sache also so aus:

10 **DATA** JOSEPHINE,22443

Beim nächsten **READ** zeigt die Markierung auf XANTIPPE, danach auf 69888 usw.

> Der Variablentyp (numerisch oder String) des **READ**-Befehls muß mit der Art der zu lesenden Information übereinstimmen.

Das heißt: Das erste **READ** bezieht sich auf Josephine, einen String. Also mußt Du eine Stringvariable angeben, zum Beispiel:

READ N$ ← Dollar-Zeichen für String

Das zweite **READ** aber bezieht sich auf 22443, eine Zahl, die eine numerische Variable erfordert:

READ NUMMER ← kein Dollar-Zeichen im Variablennamen

Für den Anfang schreiben wir ein Programm, das die **DATA**-Liste ausgibt:

```
10 DATA JOSEPHINE,22443
20 DATA XANTIPPE,69888
30 DATA KLARA,71450
40 DATA BERTA,22222
50 DATA ROSALINDE,93665
60 FOR K=1 TO 5
70 READ N$
80 READ NUMMER
90 PRINT N$;NUMMER
100 NEXT K
```

vorhergehende DATA-Liste

Sehen wir uns an, wie es arbeitet. Die Pfeile sollen die Auswirkungen der **READ**s verdeutlichen:

Wert von K	Wert von N$ wird	Wert der Nummer	Markierung zeigt auf
(vor dem Start)	-	-	JOSEPHINE
1	JOSEPHINE	-	22443
	JOSEPHINE	22443	XANTIPPE
2	XANTIPPE	22443	69888
	XANTIPPE	69888	KLARA
3	KLARA	69888	71450
	KLARA	71450	BERTA
4	BERTA	71450	22222
	BERTA	22222	ROSALINDE
5	ROSALINDE	22222	93665
	ROSALINDE	93665	DATA-Liste zu Ende, keine weiteren **READ**s erlaubt.

Siehst Du, wie die Markierung nacheinander auf jeden einzelnen Eintrag der Liste zeigt? Siehst Du, wie **READ** funktioniert?

Toms Telefonbuch

Tom Tast, der phonophile Programmierer, will sein privates Telefonverzeichnis in **DATA**s ablegen und dann ausgeben. Hier ist sein Verzeichnis:

Name	Telefonnummer
Fred Franz	667142
Amanda Bander-Gander	1234567
Lolita Nabokova	9999
Ytzak ben Nevis	434772
Igor Biva	1001001

Wie muß das entsprechende Programm aussehen?

Telefonverzeichnis

Das Ausgeben des Verzeichnisses ist kein großes Problem. Ein weiteres nützliches Programm wäre eines, das das Verzeichnis nach einem Namen durchsucht und ihn dann zusammen mit der

Telefonnummer ausgibt. Für Millie könnte dieses Programm Fragen wie „Welche Nummer hat XANTIPPE?“ beantworten und für Tom, „Ich muß sofort Lolita anrufen, aber welche Telefonnummer hat sie?“

Die Lösung ist: Alle Einträge der Liste mit **READ** auslesen und mittels **IF** mit dem gesuchten Begriff vergleichen. Wenn Eintrag und gesuchter Begriff übereinstimmen, können die entsprechenden Maßnahmen getroffen werden.

```
10 DATA FRED FRANZ,667142
20 DATA AMANDA BANDER-GANDER,1234567
30 DATA LOLITA NABOKOVA,9999
40 DATA YTZAK BEN NEVIS,434772
50 DATA IGOR BIVA,1001001
60 PRINT"ZU SUCHENDER NAME?"
70 INPUT SNAME$
80 PRINT"ICH SUCHE NACH ";SNAME$
90 FOR K=1 TO 5
100 READ NAME$
110 READ NUMMER
120 IF NAME$=SNAME$ THEN PRINT SNAME$;"
GEFUNDEN. DIE NUMMER IST";NUMMER
130 NEXT K
```

Abkürzungen

(a) Du kannst mehrere Einträge in langen Zeilen zusammenfassen und die **DATA**s über das ganze Programm verteilen. Der Computer verbindet alle **DATA**-Zeilen zu einer Liste.

(b) Du kannst mehrere Einträge mit einem Befehl lesen, zum Beispiel:

READ NAME$,NUMMER

Die Markierung springt um die entsprechende Zahl von gelesenen Einträgen weiter.

Abkürzungen

Vereinfache mit Hilfe von Methode (b) alle Programme dieses Kapitels.

RESTORE

Ich habe oben gesagt, daß keine weiteren **READ**s mehr erlaubt sind, wenn die Markierung das Ende der **DATA**-Liste erreicht hat.

Das stimmt; wenn der Computer nach dem letzten Eintrag auf ein weiteres **READ** stößt, unterbricht er die Programmausführung und gibt folgende Fehlermeldung aus:

OUT OF DATA

(Etwa: Die Daten sind mir ausgegangen.) Aber Du mußt vielleicht die **DATA**-Liste mehrere Male durcharbeiten, um zum Beispiel mehrere Einträge nacheinander herauszusuchen.

Der Befehl

RESTORE

setzt die Markierung auf den Anfang der Liste zurück. Jetzt kann sie erneut durchsucht werden. (Nebenbei bemerkt: Es gibt auch eine [RESTORE]-Taste, diese bewirkt aber etwas anderes - gib **RESTORE** wie jeden anderen BASIC-Befehl ein.) Hier ein Testprogramm:

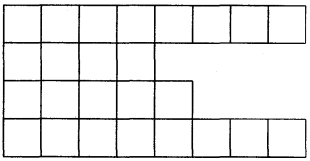
```
10 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,
   N,O,P,Q,R,S,T,U,V,W,X,Y,Z
20 FOR K=1 TO 26
30 READ X$
40 PRINT X$;
```

```
50 NEXT K
60 RESTORE
70 GOTO 20
```

Probier's mal; lösche dann Zeile 60 und versuche es erneut. Siehst Du den Unterschied?

Compucrostic

Setze die waagrechten Wörter ein und ermittle die senkrechten BASIC-Befehle.



Schienenfahrzeug
Zeitalter
Schwester eines Elternteils
Grundrechenart

Proknatakadskis Problem

Genosse Sergei Proknatakadski, der andere russische Spion, sendet eine verschlüsselte Botschaft nach Leningrad. Sein Code ändert die Buchstaben des Alphabets wie folgt:

Original	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Code	H A M E R N D S I C K L B F G J O P Q T U V W X Y Z

(a) Schreibe ein Programm, bei dem Genosse Proknatakadski seine Botschaft Buchstabe für Buchstabe eingeben kann und das die verschlüsselte Version ausgibt.
(b) Schreibe ein Programm, das solche Botschaften wieder *entschlüsselt.*

Antworten

Toms Telefonbuch

```
10 DATA FRED FRANZ,667142
20 DATA AMANDA BANDER-GANDER,1234567
30 DATA LOLITA NABOKOVA,9999
40 DATA YTZAK BEN NEVIS,434772
50 DATA IGOR BIVA,1001001
100 PRINT"TOMS TELEFONBUCH"
110 PRINT
120 FOR T=1 TO 5
130 READ X$,Y
140 PRINT X$;"    ";Y
150 NEXT T
```

Abkürzungen

Verdatete Kühe: Lösche die Zeilen 70 und 80 und ersetze sie durch:

70 **READ** N$,NUMMER

Telefonverzeichnis: Lösche die Zeilen 100 und 110 und ersetze sie durch:

100 **READ** NAME$,NUMMER

Compucrostic

DRAISINE
AERA
TANTE
ADDITION

DATA **READ**

Die **DATA**-Liste ist für beide Programme identisch: Sie wird auf zwei Zeilen verteilt.

```
10 DATA A,H,B,A,C,M,D,E,E,R,F,N,G,D,H,S
20 DATA I,I,J,C,K,K,L,L,M,B,N,F,O,G,P,J
30 DATA Q,O,R,P,S,Q,T,T,U,U,V,V,W,W,X,X,
Y,Y,Z,Z
```

Um eine Botschaft zu verschlüsseln, füge diese Zeilen an:

```
100 PRINT"NAECHSTER BUCHSTABE DER NACHRI
CHT?"
110 INPUT M$
120 RESTORE
130 FOR N=1 TO 26
140 READ A$,C$
150 IF A$=M$ THEN PRINT C$;
160 NEXT N
170 GOTO 110
```

Zum Entschlüsseln mußt Du nur Zeile 150 ändern zu:

150 **IF** C$=M$ **THEN PRINT** A$;

Die Eingabeschleife ist endlos. Du kannst eine "Ende"-Abfrage einfügen (siehe Kapitel 5):

165 **IF** M$="*" **THEN STOP**

die das Programm beendet, wenn Du "*" eingibst.

4

Erweitertes Entwanzen

In *Buch 1* haben wir uns mit Syntax- und Laufzeitfehlern beschäftigt. Diese beiden Fehlerarten machen keinen Hehl aus ihrer Anwesenheit, denn sie bringen das Programm zur Unterbrechung. Dennoch sind sie nicht immer einfach aufzuspüren. Es gibt jedoch noch gefährlichere Fehler, die das Programm zwar nicht unterbrechen, aber dafür sorgen, daß es etwas falsch macht.

Manchmal kann man diese Fehler durch ausgiebiges Starren auf das Programmlisting aufdecken, für gewöhnlich bekommt man aber nur Kopfschmerzen davon. Da hilft ein klein wenig Detektivarbeit weiter; es gibt nämlich einige sehr nützliche Tricks, die durch kleine Veränderungen des Programms zeigen, was es wirklich macht.

Testzeilen

Magarete Mausschlau entwickelt ein Programm, mit dessen Hilfe jeder feststellen kann, was es an einem bestimmten Tag zu essen gibt. So weit ist sie gekommen:

```
10 DATA MON,SPAGHETTI,DIE,EINTOPF,MIT,KO
HLAUFLAUF,DON,SCHNITZEL
20 DATA FRE,FISCH,SAM,STEAK,SON,BRATEN
30 PRINT"GEWUENSCHTER TAG"
40 INPUT GTAG$
50 FOR T=1 TO 6
60 READ TAG$,SPEISE$
70 IF TAG$=GTAG$ THEN PRINT"ES GIBT ";SP
EISE$
80 NEXT T
```

Bei einigen Probeläufen stellt sie zufrieden fest, daß es für die Tage MON und DIE gut funktioniert. Dann aber versucht sie es mit SON und - nichts wird ausgegeben. Was geht schief?

* * * * * *

Holmes richtete sich irritiert auf. „Ja, Watson?"

„Es ist soeben ein Telegramm von einer Madame Mausschlau für Sie angekommen, Holmes."

„Noch eine Wanze, Watson?"

„In der Tat, Holmes."

Holmes nahm mir das Telegramm aus der Hand. „Lassen Sie mal sehen, Watson. Hmmmmm ... Ich frage mich, warum sie am Samstag Steak ißt ...? Kein Grund ... Aha! Da das INPUT das Problem verursacht, folgere ich daraus, daß etwas NACH Zeile 40, in der das INPUT steht, nicht in Ordnung ist. Ich vermute, die Schleife, Watson."

„Ich wollte gerade dasselbe sagen, Holmes."

„Zweifellos. Ich habe schon einen Verdacht, was des Rätsels Lösung ist; um aber die letzten Zweifel zu zerstreuen, werden wir einen kleinen Test durchführen."

„Ich hole sofort den Puder für Fingerabdrücke, Holmes."

„Nein, nein, Watson, Sie altertümlicher Possenreißer. Nicht DIESEN Test. Diesen hier!"

Holmes wandte sich dem Terminal an seinem Bett zu und fügte eine Zeile ein:

```
65 PRINT T;TAG$
```

„Das sollte sehr aufschlußreich sein", sagte er.

„Wie das, Holmes?", krächzte ich.

„Es zeigt uns genau, was sich in der Schleife abspielt, Watson. Vor allem, welche Tage gelesen werden."

„Teuflisch, Holmes."

Holmes seufzte und startete das Programm. Auf dem Bildschirm erschienen die Worte GEWUENSCHTER TAG. Holmes tippte den problematischen Tag ein:

```
SON
```

Ich betrachtete erwartungsvoll den sich füllenden Bildschirm.

```
1 MON
2 DIE
3 MIT
4 DON
5 FRE
6 SAM
```

und . . . Ende.

„Aber -", ich schnappte nach Luft. „Wo zum Teufel ist SON?"

„Er wurde nie gelesen, Watson. Die Schleife kommt nur bis T=6, und das ist SAM. Damit haben wir den Schuldigen -"

„Der Endwert der Schleife!", rief ich aus. „Es müßte 7 sein!"

„Exakt, Watson."

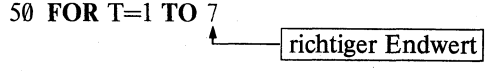
* * * * * *

Das ist eine Methode, um herauszufinden, wo das Programm fehlerhaft ist. Füge *Testzeilen* ein, die die momentanen Werte der Variablen ausgeben. So kannst Du leicht die programminternen Abläufe verfolgen. Der Computer selbst zeigt Dir, was falsch ist.

Wenn die eingefügten Testzeilen nicht weiterhelfen, lösche sie und versuche es an anderen

Stellen. Nachdem Du den Fehler gefunden und korrigiert hast, *löschst Du alle Testzeilen.*

Magarete Mausschlau machte in Zeile 50 einen Fehler. Sie hätte lauten müssen:



Korrigiere diese Zeile, lösche Zeile 65, die Testzeile, und starte das Programm erneut. Du wirst feststellen, daß es jetzt für alle sieben Wochentage einwandfrei arbeitet. (Vergiß nicht, die Dreibuchstaben-Abkürzung einzugeben: MON, DIE, usw.)

Spuren

Ein häufig anzutreffender Fehler ist eine falsche Sprungadresse. Martin Mausschlau, der mystische Mathematiker, hat ein Computerspiel entworfen. Der Computer wählt eine ganze Zahl zwischen 1 und 100. Der Benutzer des Programms muß diese Zahl erraten und erfährt, ob er zu hoch oder zu niedrig getippt hat. Wenn er richtig geraten hat, teilt ihm der Computer dies mit.

Das ist sein Programm:

```
10 DATA 54,72,66,98,4
20 READ CNUMMER
30 PRINT"WELCHE ZAHL RATEN SIE?"
40 INPUT GNUMMER
50 IF GNUMMER>=CNUMMER THEN PRINT"ZU GRO
SS"
60 IF GNUMMER>=CNUMMER THEN GOTO 30
70 IF GNUMMER=CNUMMER THEN GOTO 100
80 IF GNUMMER<CNUMMER THEN PRINT"ZU KLEI
N"
90 IF GNUMMER<CNUMMER THEN GOTO 30
```

```
100 PRINT"GUT GEMACHT!"
110 PRINT"WOLLEN SIE NOCH EINMAL SPIELEN
?"
120 INPUT F$
130 IF F$="JA" THEN GOTO 20
```

Damit sind fünf Spielrunden möglich. Für weitere Spiele füge der DATA-Liste in Zeile 10 neue Zahlen an.

Alles schön und gut, nur – es funktioniert nicht.

Beim Austesten des Programms stellte Martin folgendes fest:

```
WELCHE ZAHL RATEN SIE?
? 43
ZU KLEIN
WELCHE ZAHL RATEN SIE?
? 55
ZU GROSS
WELCHE ZAHL RATEN SIE?
? 53
ZU KLEIN
WELCHE ZAHL RATEN SIE?
? 54
ZU GROSS
WELCHE ZAHL RATEN SIE?
?
```

„Eine ganze Zahl ZWISCHEN 53 und 54", sagte Martin zu sich. Also machte er sich daran, eine völlig neue Art von Arithmetik zu entwickeln, die Theorie der hyperintangiblen exotischen Numerationsmodule. Als auch das nichts half, kam er zu dem Schluß, daß irgendwo im Programm ein Fehler stecken mußte.

* * * * * *

„Holmes, ich glaube, ich habe eine Idee!"

„Gute Arbeit, Watson, gute Arbeit."

„Ich vermute, das Programm arbeitet fehlerhaft, Holmes."

Holmes schlug sich mit der flachen Hand auf die Stirn, eine nervöse Geste, die ich in der letzten Zeit oft an ihm beobachten konnte. Ich fuhr fort: „Es scheint, daß es nicht richtig springt, Holmes."

F „In der Tat, Watson. Und was schlagen Sie vor, um diese vage Vermutung zu belegen?"

Auf seine unerwartete Frage senkte ich meinen Blick. „Ich habe nicht die leiseste Ahnung, Holmes."

Der große Mann konzentrierte sich stirnrunzelnd. Oder waren es vielleicht plötzliche Schmerzen?

„Ich schlage vor, wir legen eine Spur durch das Programm, Watson." Ich gebe meine Bestürzung zu. Wollte Holmes etwa Schnitzeljagd spielen?

„Wir müssen den Computer überreden, uns mitzuteilen, wohin er springt, Watson. Mit Hilfe geeigneter Programmzeilen."

„Aha! Wirklich schlau, Holmes!"

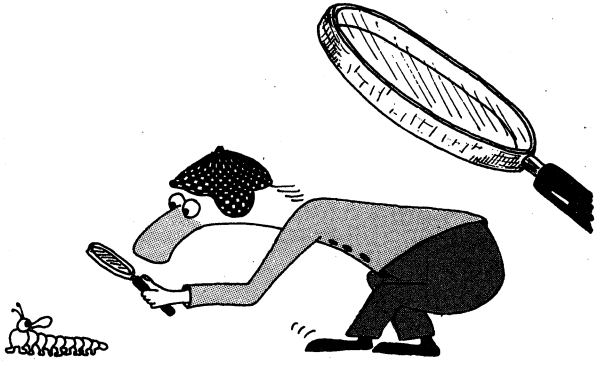
„Wollen mal sehen ... Die Sprünge lauten **GOTO** 30, **GOTO** 100 und **GOTO** 20. Ich schlage vor, wir fügen Zeilen ein, die uns mitteilen, welche Sprünge ausgeführt worden sind."

„Aber wie, Holmes?" Er sagte nichts, stattdessen huschten seine Finger über die Tastatur:

```
 21 PRINT"20 AUSGEFUEHRT"
 31 PRINT"30 AUSGEFUEHRT"
101 PRINT"100 AUSGEFUEHRT"
```

„Ach so!" Ich holte voller Bewunderung tief Luft. „Jetzt verstehe ich alles! Wenn er zu Zeile 20 springt, dann führt er auch Zeile 20 aus - einen

einfachen **READ**-Befehl –, und dann macht er mit Zeile 21, der Spur, weiter! Also teilt uns Zeile 21 mit, daß Zeile 20 ausgeführt wurde! Und die anderen Sprünge wurden mit der gleichen Spur versehen! Holmes, das ist phantastisch!"



Holmes rümpfte die Nase – bekam er etwa eine Erkältung? Er tippte **RUN.** Ich betrachtete die Wörter, die auf dem antiken Fernseher ausgegeben wurden.

```
20 AUSGEFUEHRT
WELCHE ZAHL RATEN SIE?
30 AUSGEFUEHRT
? 43
ZU KLEIN
WELCHE ZAHL RATEN SIE?
30 AUSGEFUEHRT
? 55
ZU GROSS
WELCHE ZAHL RATEN SIE?
30 AUSGEFUEHRT
? 53
ZU KLEIN
```

```
WELCHE ZAHL RATEN SIE?
30 AUSGEFUEHRT
? 54
ZU GROSS
WELCHE ZAHL RATEN SIE?
30 AUSGEFUEHRT
?
```

„Holmes, er springt *nie* zu Zeile 100. Aber er *muß,* um das Spiel zu beenden!"

Holmes zeigte lediglich die Spur eines Lächelns.

„Aber – verdammt noch mal, Holmes – Zeile 70 *zwingt* ihn, zu Zeile 100 zu springen! Sehen Sie! **IF** GNUMMER=CNUMMER **THEN GOTO** 100! Wieso springt er also nicht dahin, Holmes?"

Holmes runzelte die Stirn. Ich starrte mit Schrecken auf das Programm. „Er muß, er muß, er MUSS! Er –" Holmes unterbrach meinen Gedankenstrom.

„Es sei denn . . .", sagte er ruhig.

„Es sei denn, *was?*", rief ich in Qualen.

„Es sei denn, er kommt nie zu Zeile 70, Watson."

Ich fühlte mich, als hätte sich unter meinen Füßen ein gähnender Abgrund aufgetan. „Mein Gott, Holmes, das ist es! Natürlich! Ich wäre nie –" Aber Holmes tippte erneut:

```
69 IF GNUMMER=CNUMMER THEN PRINT"70
   AUSGEFUEHRT"
```

Ich starrte auf das Listing. „Warum 69, Holmes? Warum nicht 71?"

Holmes seufzte – eine weitere Eigenart, die ich in der letzten Zeit oft an ihm beobachten konnte.

„Watson, wenn jemals GNUMMER=CNUMMER wird und er Zeile 70 erreicht – wie er sollte –, dann springt er doch sofort zu Zeile 100. Zeile 71

würde also niemals ausgeführt. Zeile 69 hingegen ist für diese Aufgabe bestens geeignet."

Dieser Mann ist ein Genie! Ich war stolz, dieselbe Luft mit ihm atmen zu dürfen. Außer er rauchte gerade seine eigenartig riechende Pfeife.

Holmes startete das Programm erneut. Die Ausgabe ähnelte der ersten sehr stark. In der Tat, die Ausgabe war mit der ersten *identisch!* Zeile 70 wurde nie erreicht!

Holmes war eine zeitlang in Gedanken versunken. „Wir wissen aber", grübelte er, „daß die Zeilen 50 und 80 ausgeführt werden, weil *zu klein* und *zu groß* ausgegeben wird. Ich frage mich . . ."

Ich wartete mit verhaltenem Atem.

„Wir wollen uns Zeile 50 und 60 einmal näher ansehen, Watson. Was passiert, wenn wir die richtige Zahl, 54, eingeben? Dann ist sowohl GNUMMER als auch CNUMMER 54. Die Bedingung

GNUMMER>CNUMMER

ist also nicht erfüllt. Und dann . . ."

„Macht er mit Zeile 70 weiter", sagte ich. „Aber wir haben doch bereits festgestellt, daß er nie zu Zeile 70 kommt, Holmes."

„Wenn man das Unmögliche ausgeschaltet hat, Watson, dann muß das, was übrigbleibt - gleichgültig, wie unwahrscheinlich es ist - die Wahrheit sein."

„Sie meinen, der Computer ist zusammengebrochen?"

„Nein, Watson, das meine ich ganz bestimmt nicht! Wenn GNUMMER gleich CNUMMER ist, dann . . . aaaaahhhhhh!"

„Was ist, Holmes?"

„Unsere Analyse war falsch. Die Zeilen 50 und 60 sagen NICHT:

IF GNUMMER>CNUMMER...

Sie sagen:

IF GNUMMER>=CNUMMER...

„Ja, das ist es."

Ich verstand überhaupt nichts. „Kann eine so kleine Veränderung für einen so großen Fehler verantwortlich sein, Holmes?"

„Oh, in der Tat. Das ist charakteristisch, in der Tat. Sehen Sie, Watson, wenn wir Zeile 50 und 60 zu

IF GNUMMER>CNUMMER...

verändern und dann für CNUMMER die richtige Zahl 54 eingeben, wird Zeile 50 FALSCH. Wir fahren mit Zeile 60 fort ... auch falsch. Und dann ..."

„Zu Zeile 70! Und dann zu Zeile 100!"

„Exakt, Watson."

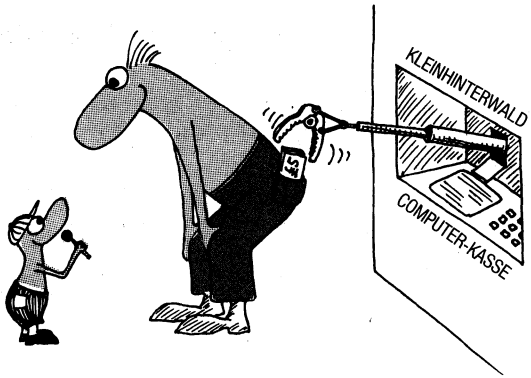
Holmes veränderte Zeile 50 und 60 und entfernte die Spuren seiner Nachforschungen. Das Programm arbeitete einwandfrei. Ich fragte mich kurz, wie er es ohne meine Hilfe hätte schaffen können, war aber zu bescheiden, den Gedanken auszusprechen.

Kleinhinterwald schlägt wieder zu

Ralf Raffke will für seine besten Kunden einen computergesteuerten Bankschalter einrichten. Er hat ein Programm geschrieben, das seinen Kunden erlaubt, ihren vorherigen Kontostand, eine Liste aller Auszahlungen und Einzahlungen, einzugeben und den neuen Kontostand abzulesen.

Er ist auf einen Trick besonders stolz, den er in den Zeilen 90 und 130 verwendet: Mit Hilfe der sinnlosen Eingabe 0 kann man eine endlose Eingabeschleife verlassen. Diese Technik wird in Kapitel 5 noch einmal ausführlich erklärt.

Weniger stolz ist er auf die Leistungen des Programms im täglichen Einsatz, da es fast nie die richtigen Antworten ausgibt. Und, was am schlimmsten ist, die Fehler wirken sich manchmal zugunsten der Kunden aus. Also hat er den Verzweifelten Volker, den Fehlerfinder (Dich!), kommen lassen, um das Programm zu überarbeiten. Kannst Du?




```
10 PRINT"SPARKASSE KLEINHINTERWALD"
20 PRINT"VORHERIGEN KONTOSTAND EINGEBEN"
30 INPUT GUTHABEN
40 LET VGUTHABEN=GUTHABEN
50 PRINT"AUSZAHLUNGEN EINZELN EINGEBEN"
60 INPUT A
70 LET VGUTHABEN=VGUTHABEN-A
80 GOTO 30
90 IF A=0 THEN GOTO 100
100 PRINT"EINZAHLUNGEN EINZELN EINGEBEN"
110 INPUT E
120 LET VGUTHABEN=VGUTHABEN+A
```

```
130 IF E=0 THEN GOTO 150
140 GOTO 110
150 LET KSTND=GUTHABEN
160 PRINT"KONTOSTAND IST";
170 PRINT KSTND
```

Testzeilen zeigen Dir, was sich verändert. *Spuren* zeigen Dir, wohin der Computer springt.

Antworten

Kleinhinterwald schlägt wieder zu

Die Zeilen 70, 80, 90 und 120 sind fehlerhaft. Sie müssen lauten:

70 **LET** GUTHABEN=GUTHABEN–A
80 **IF** A=0 **THEN GOTO** 100
90 **GOTO** 60
120 **LET** GUTHABEN=GUTHABEN+E

5
Neues von den Strings

Viele Leute stellen sich Computer als Maschinen vor, die ZAHLEN verarbeiten und komplizierte Rechnungen ausführen. Diese Fähigkeiten des Computers bezeichnet man in der Fachwelt als „number-crunching" (Zahlen-Verdauung); auf diesem Gebiet sind Computer sicherlich unschlagbar. Aber nur Wissenschaftler sind wirklich auf diese fast überdimensionierten Fähigkeiten angewiesen. Computer können noch vieles andere: Graphiken zeichnen, Datensätze für Geschäftsleute verwalten, Maschinen steuern oder Briefe schreiben.

Du hast schon gesehen: Einige Programme verdauen keine Zahlen. Es gibt Programme, die Daten darstellen, und solche, die sie verwalten. Programme wie diese sind möglich, weil ein Computer alle Arten von Symbolen verdauen kann, nicht nur Zahlen. Er verarbeitet Informationen: Fakten, Zahlen, alles, was durch Symbole dargestellt werden kann. Der Plan einer Brücke, der Text zu „Herr der Ringe", die Noten einer Beethoven-Symphonie oder eine Anzeige für ein Erfrischungsgetränk.

Viele Branchen verwenden jetzt den Computer als „intelligente Schreibmaschine" oder Textverarbeitungssystem. Hier wird der Computer zur Verdauung von Worten und Buchstaben eingesetzt. Er kann Rechtschreibfehler verbessern und jederzeit einen sauberen Ausdruck des bis dahin eingegebenen Textes erzeugen.

Das grundsätzliche Prinzip, auf dem die Bearbeitung solcher Problembereiche beruht, ist die Idee des *Strings*. Erinnern wir uns an Buch 1:

> Ein *String* ist eine Reihe von zusammengehörigen Zeichen, also eine Zeichenkette. Stringvariablen müssen durch ein "$" am Ende ihres Namens gekennzeichnet sein.

Wenn Du mit Hilfe des LET-Befehls eine Stringvariable definierst, mußt Du den Wert der Stringvariablen in Anführungszeichen setzen:

```
LET A$ = "FRED"
LET KATZE$ = "FELIX"
LET G$ = "BACH"
```

usw.

> Die Anführungszeichen, die einen String umschließen, markieren Anfang und Ende des Strings. Sie sind nicht Teil des Strings.

Die Anführungszeichen ähneln Buchstützen, die Bücher an ihrem Platz halten, aber selbst keine Bücher sind, klar? Anführungszeichen halten den String an seinem Platz, sind aber keine Zeichen *innerhalb* des Strings.

Wie lang ist ein String?

Um die Länge eines Strings, das heißt die Anzahl der Zeichen, die er enthält, zu erfahren, verwendest Du das Befehlswort

LEN

in der Form

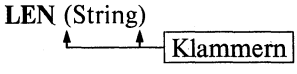
LET Variable = **LEN**(String)

Zum Beispiel:

```
LET X = LEN("FRED")
LET L2 = LEN(KATZE$)
```

In diesen Beispielen nimmt X den Wert 4 an, denn der String "FRED" enthält vier Buchstaben. L2 wird der Wert 5 zugewiesen, denn die Variable KATZE$ hat den Wert "FELIX", und der String "FELIX" enthält fünf Buchstaben.

Beachte die Klammern:



Das Wort **LEN** ist eine neue Art von Befehlswort. Es ist kein eigenständiger Befehl; folgende Zeile ist unzulässig:

```
10 LEN dies oder das
```

LEN ergibt nur einen Wert, der noch weiterverwendet werden muß. Befehlswörter wie LEN nennt man FUNKTIONEN.

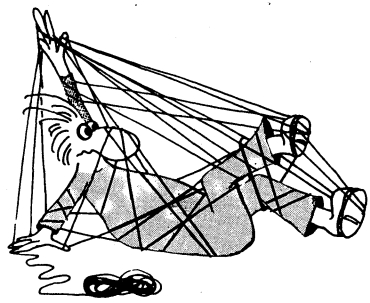
> Wenn Du eine Variable in eine *Funktion* einsetzt, bekommst Du einen Wert, der von dem Wert der Variablen über eine bestimmte Regel abhängig ist.

In diesem Fall lautet die Regel „zähle, wieviele Zeichen im String enthalten sind". Wenn Du einen String (oder eine Stringvariable) in die Lücke zwischen den Klammern bei LEN () einsetzt, dann ist das Ergebnis eine Zahl.

Enthält ein String LEERZEICHEN, so werden diese durch **LEN** ebenfalls gezählt. **LEN**("HARRY HIRSCH") ergibt zum Beispiel 12, denn auch das LEERZEICHEN gilt als Zeichen.

Bei diesem Beispielprogramm gibst Du einen String ein; das Programm sagt Dir dann, wie lang dieser ist.

```
10 PRINT "STRING EINGEBEN!"
20 INPUT STRING$
30 PRINT"IHR STRING ENTHAELT";LEN
   (STRING$);"ZEICHEN"
```



Strings verketten

Das Zeichen

+

kennen wir schon in seiner Bedeutung „addiere zwei Zahlen". In Verbindung mit Strings hat es jedoch eine andere Wirkung. Der Ausdruck

A$+B$

bedeutet „hänge B$ ans Ende von A$ an". So ergibt

"ROT" + "KOHL" "ROTKOHL"

Die gängige Bezeichnung für diese String-Addition lautet VERKETTUNG von Strings.

Übungsaufgaben

1. Was ergeben die folgenden Verkettungen?

   a) "ZAHN"+"PASTA"
   b) "TAN"+"NE"
   c) "BU"+"EGELN"
   d) "GRI"+"ECH"+"ISCH"
   f) "X"+"Y"+"L"+"O"+"PHON"

2. Finde alle möglichen String-Verkettungen, die den String

   "APFEL"

   zum Ergebnis haben.
3. Laß diese beiden Programme laufen:

   a) 10 **PRINT** "FLUSS"+"PIRATEN"
   b) 20 **PRINT** "PIRATEN"+"FLUSS"

   Sind die Ergebnisse gleich? Wie lauten sie?
4. Finde zwei Strings A$ und B$, so daß

   A$+B$="RABE"
   B$+A$="ABER"

Komputer machen keine Rechtschreibfehler

Peter Wirr, Chef-Manager der Unclear Selectronics Deutschland AG, ist Verfasser eines hundert

Seiten starken Dokuments, das die Zukunftspläne der Firma zusammenfaßt. Leider muß er am späten Nachmittag bemerken, daß er ein Wort immer wieder falsch geschrieben hat. Niemand hatte ihn bis dahin darauf aufmerksam gemacht, daß KOMPUTER einen Fehler enthält.

Glücklicherweise enthält seine Programmbibliothek ein Komputer-Programm, das die Sache noch retten kann.

```
10 INPUTW$
20 FORT=1TO20
30 PRINT" ";
40 NEXTT
50 IFW$<>"KOMPUTER"THENPRINTW$
60 IFW$="KOMPUTER"THENPRINT"COMPUTER"
70 GOTO10
```

Mit diesem Programm kann er das Dokument Wort für Wort eingeben; das Programm korrigiert dabei den Fehler und druckt die korrigierte Version am rechten Rand des Bildschirms aus. (Wenn Du noch etwas bastelst, kannst Du die Eingabe-Meldungen unterdrücken, aber das geht über den Rahmen dieses Buches hinaus.)

Versuche es mit dem ersten Satz des Wirr-Dokuments:

DIE NEUE ABTEILUNG FUER KOMPUTER DER UNCLEAR SELECTRONICS HAT SOEBEN DAS PROJEKT ZZUB HOME KOMPUTER SYSTEM ABGESCHLOSSEN.

Du hast wahrscheinlich das neue Symbol im Programmtext bemerkt:

< >

Dies bedeutet „ungleich" und wird erzeugt, indem Du

eingibst.

Ich möchte Dir nicht erzählen, daß echte Textverarbeitungssysteme so schwerfällig und simpel aufgebaut sind. Du mußt natürlich nicht ein Wort nach dem anderen, einzeln, eingeben! Der Text wird von Anfang an auf dem Computer geschrieben worden sein, so daß er sich schon im Speicher befindet. Der Computer kann dann auf einen Befehl den Text durchsuchen und jedes Wort „KOMPUTER" zu „COMPUTER" ändern. Das Prinzip des Austauschens aber ist dasselbe.

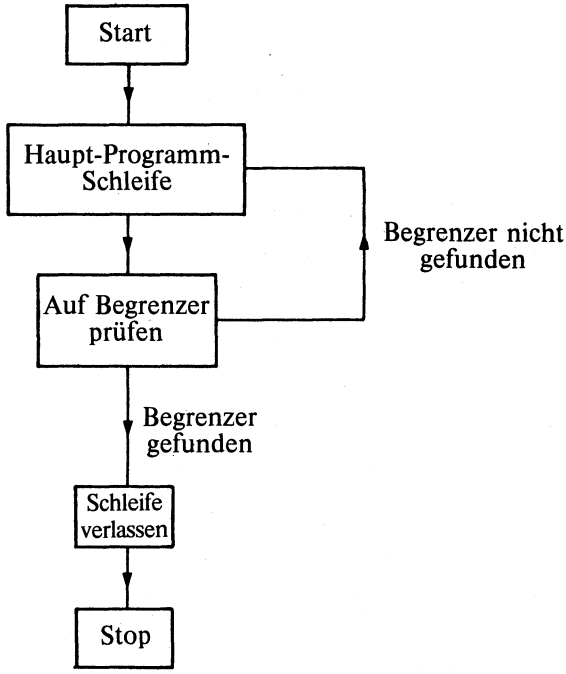
Begrenzer

Der Befehl **GOTO** 1Ø läßt das obige Programm immer weiterarbeiten. Du müßtest den Computer ausschalten, um es abzubrechen. Um jederzeit abbrechen zu können, kannst Du Dir etwas bei Ralf Raffke abgucken und einen „Begrenzer"

verwenden. Dies ist ein beliebiger String, der in einem herkömmlichen Text nicht auftritt, zum Beispiel "XXXXX". Ändere also Zeile 70 ab zu

```
70 IF W$ <> "XXXXX" THEN GOTO 10
```

Wenn Du nun nicht den Begrenzer eingibst, fährt das Programm in Zeile 10 fort und fragt nach dem nächsten Wort. Gibst Du aber "XXXXX" ein, dann verläßt der Computer die Schleife und bricht ab.



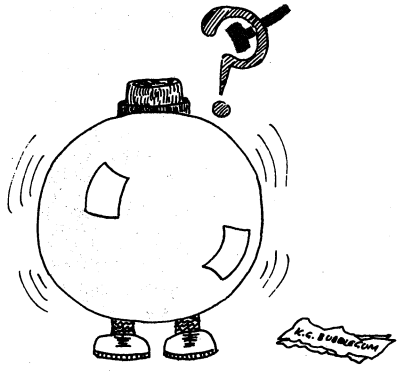
Ein *Begrenzer* ist eine spezielle Eingabe, die dem Computer signalisiert, daß er anders als üblich handeln muß. Oft soll er eine Eingabe-Schleife verlassen.

Der Spion gibt nicht auf

Genosse Proknatakadski, der Spion, hat die Geheimpläne einer amerikanischen Kaugummi-Fabrik entdeckt und schickt sie nun nach Moskau, in der Hoffnung, daß russische Wissenschaftler einen Kaugummi entwickeln können, der so groß ist, daß er die ganze Welt in die Luft sprengen kann. Er versteckt die Botschaft in einem Programm.

```
10 LETA$="UG"
20 LETB$="EN"
30 LETC$="TA"
40 LETD$="E KO"
50 LETE$="SKI"
60 LETF$="MM"
70 LETG$="G AB"
80 LETH$=" SONN"
90 LETI$="D MI"
100 LETJ$="M FL"
110 LETK$="T DE"
120 LETL$="ZE"
130 LETM$=" GRUES"
140 LETN$="KAD"
150 LETO$="SE PROKNA"
160 LETP$="PLA"
170 PRINT"STRENG GEHEIME NACHRICHT:"
180 PRINT"BETRIFFT 50-MEGATONNEN-KAUGUMM
I"
190 PRINTP$+B$+D$+F$+B$+B$+H$+C$+G$+B$+I
$+K$+J$+A$+L$+A$+M$+O$+C$+N$+E$
200 PRINT"DASWIDANIJA!"
```

Wann kommen die Pläne an? Sollte der KGB Agenten zur Bushaltestelle, zum Bahnhof oder zum Flughafen schicken, um sie abzuholen?



Der leere String

Wenn Du mit Zahlen arbeitest, ist die Zahl 0 sehr nützlich. Sie hat eine bestimmte Eigenschaft:

$$X + 0 = X = 0 + X$$

Dies gilt für JEDES X.

Es gibt einen String mit ähnlichen Eigenschaften:

$$X\$ + \text{“???”} = X\$ = \text{“???”} + X\$$$

gültig für jedes X$.

Hast Du eine Idee?

Wenn Du länger nachdenkst, kommst Du zu dem Schluß, daß es ein String sein muß, der irgendwie KEINE Zeichen enthält. Dieses ominöse Etwas nennt man den leeren String. Er wird konsequenterweise so ausgedrückt:

```
""
```

Also zwei Anführungszeichen mit nichts dazwischen. (Ein leeres Buchregal besteht aus zwei

Buchstützen mit nichts dazwischen, oder?)

Das „Compuwahl"-Programm aus Kapitel 1 speichert die Stimmen, indem es eine „laufende Gesamtsumme" für jeden Kandidaten führt - die Variablen erhalten den Anfangswert 0 und werden dann in einer Schleife, je nachdem, wie die Stimmen fallen, hochgezählt. Der leere String "" wird oft ähnlich gebraucht, wenn in einer Schleife ein komplizierter String aufgebaut werden soll. Zum Beispiel:

```
10 LET E$=""
20 LET E$=E$+"TICK"
30 PRINT E$
40 GOTO 20
```

Versuche abzuschätzen, was dieses Programm bewirkt; laß es laufen (RUN) und stelle fest, ob Du recht hast.

Eigenbau-Pavian

Das nächste Programm benutzt den leeren String "" und die Verkettung + zur Verwirklichung eines Wortspiels.

```
10 LETY$="GESTERN LIEF EIN "
20 LETB$="PAVIAN DIE STRASSE ENTLANG"
30 LETA$=""  ← leerer String
40 PRINTY$+A$+B$
50 PRINT  ← Leerzeile
60 PRINT"GEBEN SIE EIN ADJEKTIV EIN!"
70 INPUTI$
80 LETA$=I$+" "+A$
90 GOTO40
```

Laß es laufen (RUN). Immer wenn Du nach einem Adjektiv gefragt wirst, gibst Du eines ein, aber im Nominativ Singular, also:

GRUENER
EXZENTRISCHER
FEDERNDER
INTERKONTINENTALER
SCHLAUER

usw. Der Computer erzeugt eine Beschreibung Deines Pavians, indem er die Strings zusammenbaut. Wenn Du die Beispielworte eingibst, ergibt dies schließlich:

GESTERN LIEF EIN SCHLAUER INTERKONTINENTALER FEDERNDER EXZENTRISCHER GRUENER PAVIAN DIE STRASSE ENTLANG

Prüfe, wie lang der Satz maximal werden kann. (Der Computer unterbricht das Programm, wenn der String zu lang für sein Stringbehandlungssystem wird.) Ich bin so weit gekommen:

GESTERN LIEF EIN RIESIGER HAARIGER FALLSUECHTIGER BILLIGER ZERBRECHLI-

CHER UNVERSTAENDLICHER WUESTER
SCHWITZENDER UNSICHTBARER
SCHLAUER INTERKONTINENTALER
FEDERNDER EXZENTRISCHER GRUENER
PAVIAN DIE STRASSE ENTLANG
Aber Du kannst das bestimmt besser.

Antworten

Übungsaufgaben

1. a) "ZAHNPASTA"
   b) "TANNE"
   c) "BUEGELN"
   d) "GRIECHISCH"
   e) "XYLOPHON"

2. Sechzehn Möglichkeiten!
   "APFEL"
   "APFE"+"L"
   "APF"+"EL"
   "APF"+"E"+"L"
   "AP"+"FEL"
   "AP"+"FE"+"L"
   "AP"+"F"+"EL"
   "AP"+"F"+"E"+"L"
   "A"+"PFEL"
   "A"+"PFE"+"L"
   "A"+"PF"+"EL"
   "A"+"PF"+"E"+"L"
   "A"+"P"+"FEL"
   "A"+"P"+"FE"+"L"
   "A"+"P"+"F"+"EL"
   "A"+"P"+"F"+"E"+"L"

3. Nein. a) ergibt "FLUSSPIRATEN" b) ergibt "PIRATENFLUSS"

4. A$="R" B$="ABE"

Der Spion gibt nicht auf

Das Programm gibt folgende Meldung aus:

PLAENE KOMMEN SONNTAGABEND MIT DEM FLUGZEUG GRUESSE PROKNATA-KADSKI

Die Pläne kommen also Sonntagabend an, und der KGB sollte seine Leute zum Flugplatz schicken.

Der leere String

Es druckt etwa dies aus:

TICK
TICKTICK
TICKTICKTICK
TICKTICKTICKTICK

bis kein Platz mehr da ist oder Du genug hast und Deinen Computer durch das Drücken des Netzschalters erlöst.

6
Einmal Feld-Herr

Es ist fünf Uhr morgens an einem kalten Wintertag. Ich liege in meinem Bett und höre, wie meine Nachbarn das Flachdach ihrer Garage erklimmen. Gelegentlich fällt der Lichtstrahl einer Taschenlampe durch einen Schlitz zwischen den Vorhängen in mein Schlafzimmer.

„Gib mir den Regenmesser, Marta."

„Einen Moment, Max. Ich glaube, mein Fuß steckt fest. Und das Papier des Barometers hat sich um meinen Hals gewickelt."

„Der Windmesser ist verbogen; das muß der Sturm gewesen sein."

„Oder die Katze der alten Schröckschraup ... wieder auf der Jagd nach Tauben."

„Na los, gib mir schon den Regenmesser!"

„Es ist der verfluchte Regenmesser, in dem mein Fuß steckt, du Tölpel. Ich hab' Dir doch tausend Mal gesagt, daß Du das Ding nicht dahin legen sollst, wo jemand hintreten könnte!"

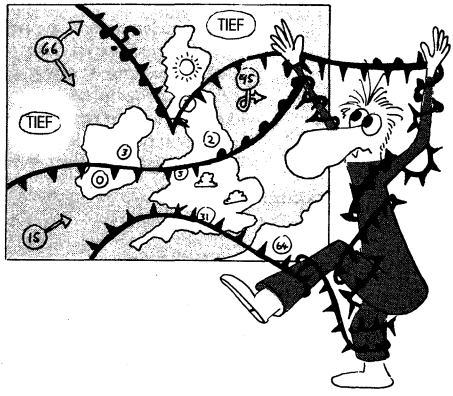
„Ja, aber hier oben ist nicht viel Platz; was ist eigentlich mit dem Sonargerät, den zwei alten Fahrrädern und dem Kasten toter Begonien? Wo sonst, stellst Du Dir vor, hätte ich -"

Es sind Marta und Max Monsun, die martialischen Meteorologen. Jeden Morgen besteigen sie ihre Garage und messen das Wetter.

Plötzlich höre ich ein lautes Platschen. Ich vermute, Max ist über die Fahrräder gestolpert und in der neben der Garage stehenden Regentonne gelandet. Ich hoffe nur, er hat das Barometer nicht mit in die Tiefe gerissen. Man sagt, ein fallendes Barometer ist ein Zeichen für schlechtes Wetter.

Es muß doch einen einfacheren Weg geben, das Wetter aufzuzeichnen. Warum AUTOMATISIEREN sie ihre Wetterstation nicht? Dann könnten sie mit einem Computer die aufgezeichneten Daten auswerten. Ich öffne das Fenster und schlage ihnen meine Idee vor. Auf verdutztes Schweigen folgt ein empörter Schrei von Marta Monsun: „Sie und Ihre verfluchten Computer! Biep-Biep-Biep die ganze Nacht lang! Zisch! Pauhhh! Kabumm! Sie bringen die ganze Nachbarschaft um ihre verdiente Nachtruhe!"

Ich ducke mich, als eine alte Rübe nur wenige Zentimeter von meiner Nase entfernt die Wand trifft. Triumphierend schließe ich das Fenster. Ich glaube, sie haben Gefallen an meiner Idee gefunden.



Datenspeicherung

Offensichtlich muß ich die Sache noch einmal durchdenken, bevor ich damit erneut an sie herantrete. Ein Problem ist, die aufgezeichneten Daten so zu speichern, daß der Computer sie weiterverarbeiten kann. Dafür KÖNNTE ich **DATA**-Listen verwenden, diese sind aber nicht

darauf ausgelegt, einfach verändert zu werden. Ich glaube, ich nehme ein *Feld.*

Ein Feld ist eine numerierte Liste. Es hat einen Namen wie eine Variable und zusätzlich noch eine Zahl, die die Position auf der Liste bestimmt.

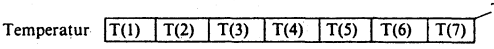
Das sind die Daten einer Woche aus der Monsun-Wetterstation:

	MON	DIE	MIT	DON	FRE	SAM	SON
Temperatur (°C)	15	3	0	8	7	2	11
Niederschläge (cm)	0	1	1	0	2	3	0
Sonnenschein (Std)	8	6	5	2	0	1	9
Windgeschwindigkeit (km/h)	24	93	11	14	6	2	18

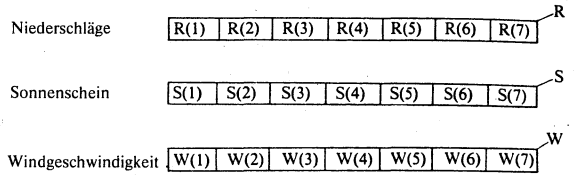
Eine Woche hat sieben Tage; ich könnte sie also von eins bis sieben numerieren und ein Feld der Größe 7 verwenden, um die Werte abzulegen. Ich brauche vier Felder: eines für die Temperatur, eines für die Niederschläge, eines für den Sonnenschein und eines für die Windgeschwindigkeit. Um sie mir leichter merken zu können, nenne ich sie T, N, S und W. Die Zahlen, die zeigen, an welcher Stelle der Liste man sich befindet, schreiben wir hinter den Namen in Klammern. Zum Beispiel:

> T(3) ist die Temperatur am Tag 3 (Mit), also 0.
> N(6) sind die Niederschläge am Tag 6 (Sam), also 3.

Dann werden alle Daten im Computer auf diese Art gespeichert:

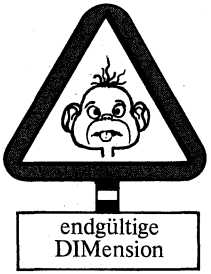


Das ist ein Feld der Größe 7; die drei anderen:



Die Frage ist nur, wie setzen wir das in BASIC um?

Ein Feld aufbauen



Um die Sache einfacher zu machen, wollen wir uns zunächst einmal nur mit einem Feld beschäftigen: dem Sonnenschein-Feld, S.

Als erstes müssen wir dem Computer mitteilen, daß er Speicherplatz für ein Feld bereitstellen soll. Das wird DIMENSIONIEREN des Feldes genannt und mit dem Befehl **DIM** bewerkstelligt. Die Programmzeile

```
10 DIM S(7)
```

weist den Computer an, ein Feld mit dem Namen S einzurichten, das Platz für 7 Zahlen hat.

Jetzt müssen wir die Zahlen in diesem Feld speichern. Hier ist eine umständliche, dafür leichter verständliche Methode:

```
20 LET S(1) = 8
30 LET S(2) = 6
40 LET S(3) = 5
50 LET S(4) = 2
60 LET S(5) = 0
70 LET S(6) = 1
80 LET S(7) = 9
```

Ein *Feld* ist eine Liste von Zahlen. Bevor man ein Feld verwenden kann, muß man dem Computer mit dem DIM-Befehl mitteilen, wie lang die Liste ist. Das wird DIMENSIONIEREN DES FELDES genannt.

Übertragen einer DATA-Liste

Eine andere Methode ist die Übertragung der Zahlen aus einer DATA-Liste in das Feld. Dort können die Zahlen dann einfach weiterverarbeitet werden. So zum Beispiel:

```
10 DATA 8, 6, 5, 2, 0, 1, 9
20 DIM S(7)
30 FOR N = 1 TO 7
40 READ X
50 LET S(N) = X
60 NEXT N
```

Du kannst sogar die Zeilen 40 und 50 durch

```
40 READ S(N)
```

ersetzen, um Zeit zu sparen.

Daten aus einem Feld beziehen

Gleichgültig, welche Methode Du verwendest, sind wir jetzt in der Lage, mit dem Feld zu arbeiten. Unsere erste Aufgabe soll sein, ein Programm zu entwickeln, das uns für einen bestimmten Tag die Menge an Sonnenschein ausgibt. Das ist ganz einfach. Gib die Zeilen 10-80 der umständlichen

Version (oder die Zeilen 10-60 der DATA-Version) ein und füge die folgenden hinzu:

```
100 PRINT"GEBEN SIE DEN TAG EIN. (1-7)"
110 INPUT TAG
120 PRINT"DIE MENGE AN SONNENSCHEIN
    WAR";
130 PRINT S(TAG)
```

Nun wollen wir auch sichergehen, daß Du verstehst, was wir hier tun. Nehmen wir an, Du willst wissen, wie lange die Sonne am Freitag geschienen hat. Freitag ist Tag Nummer 5. Also solltest Du in Zeile 110, nach der Aufforderung von Zeile 100, die Zahl 5 eingeben.

Der Computer weist der Variablen TAG den Wert 5 zu.

In Zeile 130 sieht der Computer nach, welchen Wert S(TAG) hat. Zuerst findet er den Wert von TAG, der ist 5. Jetzt weiß er, daß er unter S(5) suchen muß. Zeile 60 des „umständlichen" Programms von vorhin (oder der fünfte Eintrag der DATA-Listen-Version) teilen dem Computer mit, daß S(5) Null ist. Die Ausgabe lautet also:

```
DIE MENGE AN SONNENSCHEIN WAR 0
```

Das Feld stellt Dir sozusagen sieben verschiedene Variablen zur Verfügung: S(1), S(2), ..., S(7). Aber das ist noch nicht alles: Du kannst Dich auf jede Variable mittels S(N) beziehen, denn N kann eine Zahl von 1-7 sein. Ein Programm kann den Wert von N ändern und so mit den sieben verschiedenen Zahlen des Feldes arbeiten.

Ohne Felder könntest Du zwar auch sieben Variablen S1, S2, ..., S7 verwenden, wenn Du aber den Computer anweist, nach SN zu suchen, würde er nach einer Variablen MIT DEM NA-

MEN SN suchen. Selbst wenn er wüßte, daß N=4 ist, würde er nicht nach S4 suchen. Die Klammern um das N und die Dimensionierung des Feldes S machen erst den Unterschied aus!

Übungsaufgaben

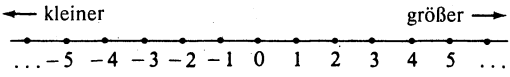
1. Wie lauten die Ergebnisse der folgenden Rechnungen? S(1) - S(7) haben die oben aufgeführten Werte.

   (a) S(1)+S(2)
   (b) S(3)–S(7)
   (c) 2*S(6)
   (d) S(3)*S(7)
   (e) S(2+2)
   (f) S(1+2+3)*S(2*3)–2*S(2*2+1)+S(28/4)

   Errechne die Ergebnisse im Kopf und prüfe sie dann mit dem Computer nach. (Wie?)
2. Finde für S(1)–S(7) ähnliche Formeln wie in 1., um die Zahlen von 0 bis 9 auszudrücken. Versuche es zuerst auf dem Papier und überprüfe es dann mit dem Computer.
3. Schreibe ein Programm, das die Windgeschwindigkeiten einer Woche in dem Feld W speichert. Der Benutzer kann die Windgeschwindigkeit eines bestimmten Tages erfragen.

Numerische Reihenfolge

Dieser Kasten soll Dich daran erinnern, daß Zahlen ihrer Größe nach sortiert werden können. Wir können also von der *größeren* und der *kleineren* von zwei Zahlen sprechen. Wenn Du Dir die Zahlen auf einer Linie angeordnet vorstellst (dem sogenannten Zahlenstrahl), dann liegen die größeren Zahlen auf der rechten und die kleineren auf der linken Seite.



Es gibt einige Standardsymbole, die der Computer zur Darstellung der Reihenfolge von Zahlen verwendet:

M=N bedeutet, daß M *gleich* N ist. (Das weißt Du natürlich schon!)

M<>N bedeutet, daß M *ungleich* N ist. (Beispiele: −2 <> 4 ist eine wahre Aussage; 2+2<>4 ist eine falsche.)

M>N bedeutet, daß M *größer als* N ist. (Beispiele: 5>4 ist wahr; 3>−4 ist wahr; −4>−5 ist wahr; −5>−4 ist falsch; 2>7 ist falsch.)

M<N bedeutet, daß M *kleiner als* N ist. (Beispiele: 2<3 ist wahr; −1<4 ist wahr; −3<−2 ist wahr; 3<2 ist falsch; −1<−99 ist falsch.) Beachte, daß M<N das gleiche bedeutet wie N > M.

M>=N bedeutet, daß M *größer oder gleich* N ist. (Beispiele: 3>=2 ist wahr; 3>=3 ist wahr; 7>=9 ist falsch.)

M<=N bedeutet, daß M *kleiner oder gleich* N ist. (Beispiele: 2<=7 ist wahr; −3<=−3 ist wahr; 5<=5 ist wahr; 6<=5 ist falsch; 999<=73 ist falsch.) Beachte, daß M<=N das gleiche bedeutet wie N>=M.

Das gleiche gilt auch für ganze und gebrochene Zahlen.

Felder verarbeiten

Das ist zwar einfach zu verstehen, aber auch nicht besonders aufregend. Wenn das alles wäre, was man mit Feldern machen könnte, wären sie die Mühe nicht wert. Aber Computer speichern nicht nur Daten; sie verarbeiten sie auch.

Was könnten die Monsuns alles wissen wollen?

1. Den Gesamtsonnenschein einer Woche.
2. Den durchschnittlichen Sonnenschein pro Tag.
3. Den Tag mit dem meisten Sonnenschein (oder die Tage, wenn es mehr als einen gibt) und die Dauer des Sonnenscheins an diesem Tag.
4. Dasselbe für den Tag des geringsten Sonnenscheins.

Es gibt noch weitere interessante Fragen:

5. An welchen Tagen der Woche schien die Sonne zwischen 1 und 5 Stunden?
6. Gab es Tage, an denen die Sonne überhaupt nicht schien? Wenn ja, welche?
7. Wie lange schien die Sonne durchschnittlich am Wochenende (Samstag und Sonntag)?
8. Schien die Sonne länger am Samstag oder am Sonntag?

Wenn die Daten erst einmal im Computer gespeichert sind, ist es ein leichtes, mit kurzen Programmen Antworten auf alle diese Fragen zu finden. Versichere Dich, daß sich NUR die Daten-Laderoutinen (Zeilen 10-80 bzw. 10-60 der beiden obenstehenden Programme) im Speicher des Computers befinden und füge dann die folgenden Zeilen an:

Gesamtsonnenschein

```
100 LET SUM=0
110 FOR N=1 TO 7
120 LET SUM=SUM+S(N)
130 NEXT N
140 PRINT"GESAMTSONNENSCHEIN
    BETRAEGT";SUM;"STUNDEN."
```

Durchschnitt

Dieses Programm ist Deine Aufgabe. Der *Durchschnitt* einer Reihe von Zahlen ist ihre Gesamtsumme, dividiert durch die Anzahl der Zahlen. Nehmen wir zum Beispiel diese Zahlen:

1, 9, 7, 2, 2, 3

Ihre Summe ist 24, da 1+9+7+2+2+3=24. Es sind sechs Zahlen. Der Durchschnitt errechnet sich also durch:

$$\text{Durchschnitt} = \frac{\text{Summe}}{\text{Anzahl}} = \frac{24}{6} = 4$$

Kannst Du das Programm „Gesamtsonnenschein" durch Ändern einer Zeile dazu bringen, den durchschnittlichen Sonnenschein pro Tag auszurechnen?

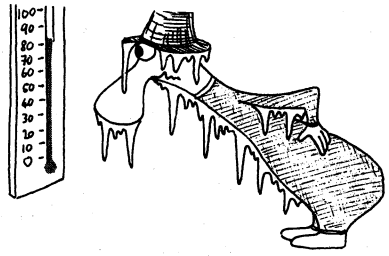
Maximaler Sonnenschein

Um das Programm einfach zu halten, wollen wir zunächst einmal nur die größte Menge an Sonnenschein ermitteln, ohne uns darum zu kümmern, an welchem Tag sie gemessen wurde. Die entscheidende Idee des Programms ist die Einführung einer Variablen, nennen wir sie MAX, die immer dann erhöht wird, wenn wir einen Wert von S(N) finden, der größer ist.

(Erinnere Dich an die Übersicht auf Seite 73.)

Das ist das Programm:

```
100 LET MAX=0
110 FOR N=1 TO 7
120 IF S(N)>MAX THEN LET MAX=S(N)
130 NEXT N
140 PRINT"DIE GROESSTE MENGE AN SON
    NENSCHEIN IST";MAX
```



Zur Probe wollen wir einen TROCKENLAUF machen (s. „Commodore 64 für Anfänger", Kap. 9).

Das ist die Trockenlauftabelle, die uns zeigt, wie sich N, S(N) und MAX innerhalb der Schleife verändern.

Zeilennummer	MAX	N	S(N)	S(N)>MAX?
100	0	—	—	
110	0	1	8	Ja
120	8	1	8	
130	8	2	6	Nein
120	8	2	6	
130	8	3	5	Nein
120	8	3	5	
130	8	4	2	Nein
120	8	4	2	
130	8	5	0	Nein
120	8	5	0	
130	8	6	1	Nein
120	8	6	1	
130	8	7	9	Ja
120	9	7	9	
130		Ende der Schleife		
140	9 ←	das gesuchte Maximum		

Für schnelle Denker

Hätte diese Methode auch funktioniert, wenn wir in Zeile 100 MAX auf 10 gesetzt hätten? Wenn ja, warum? Wenn nein, warum nicht? Welche Werte eignen sich als Startwerte für MAX, wenn diese Methode funktionieren soll?

Minimaler Sonnenschein

Noch eine Aufgabe für Dich. Überlege Dir, wie MAX funktioniert, und passe das Programm für

eine Variable MIN an, die die geringste bis jetzt gefundene Menge an Sonnenschein enthält. SEI MIT DEM STARTWERT FÜR MIN VORSICHTIG! 0 ist keine gute Idee. Erinnere Dich: A<B bedeutet „A ist kleiner als B".

An welchen Tagen?

Nachdem wir die größte Menge an Sonnenschein gefunden haben, können wir das Feld noch nach dem Tag (oder den Tagen) durchsuchen, an dem die Sonne am längsten geschienen hat. Behalte die Zeilen 10-80 (oder 10-60) und das Programm „Maximaler Sonnenschein" (Zeilen 100-140) im Speicher und füge hinzu:

```
200 FOR N=1 TO 7
210 IF S(N)=MAX THEN PRINT"AUFGETRE
    TEN AM TAG";N
220 NEXT N
```

Jetzt, da wir den Wert MAX kennen, können wir einfach den Sonnenschein jedes Tages mit MAX vergleichen und, wenn beide Werte gleich sind, den entsprechenden Tag ausgeben.

Noch einige Aufgaben

Die Programme, die die Fragen 5, 6, 7 und 8 beantworten, überlasse ich Dir.

Große Datenmengen

Du machst Dir für den Sonnenschein einer Woche natürlich nicht all diese Arbeit. Wie wär's

aber mit 1Ø Jahren? Das wären 365Ø Tage (ohne die Schaltjahre zu berücksichtigen; mit ihnen höchstens 3652). Und alles, was Du zu tun hättest, wäre die 7 in eine 365Ø zu ändern. Das Feld würde wahrscheinlich mit einer Eingabeschleife „geladen", und alle Werte würden auf einem externen Speichermedium, zum Beispiel einer Kassette oder Diskette, gespeichert. Auf diese Techniken kann ich im Moment noch nicht näher eingehen; die in diesem Kapitel vorgestellten Methoden werden sich aber bestimmt bei größeren Datenmengen auszahlen.

Es gibt auch für kleine Felder unzählige Anwendungen: Sie gehören zu den vielseitigsten Werkzeugen des Programmierers. Im Laufe dieses Buches wirst Du noch einige davon kennenlernen.

Computer-Experiment

Mit dem folgenden Experiment wollen wir veranschaulichen, wie man Informationen sammeln und automatisch verarbeiten kann. Da uns aber eine aufwendige elektronische Ausrüstung fehlt, mußt Du die Rolle der Apparate für automatische Informationsaufzeichnung spielen.

Sammle die Wetterdaten eines Monats. Nimm dafür entweder ein Thermometer und miß jeden Tag, zur selben Zeit die Außentemperatur, oder notiere die Werte, die die Zeitung zum Beispiel aus Reykjavik meldet. (Oder mogle einfach und verwende die untenstehende Datenliste.) Am Ende hast Du ungefähr 3Ø Temperaturwerte. Das folgende Programm ermittelt die durchschnittliche, die maximale und die minimale Temperatur des Monats. Es kann leicht an andere Meßreihen, wie Niederschläge, Luftdruck, Windgeschwindigkeit, usw., angepaßt werden. Die meiste Arbeit

macht das Aufzeichnen und Eingeben der Daten in den Computer.

```
10 PRINT"TEMPERATUR-ANALYSE"
20 PRINT
30 PRINT"WIEVIELE WERTE HABEN SIE AUFGEZ
EICHNET?"
40 INPUT D
50 DIM T(31)
60 PRINT"WERTE EINZELN NACHEINANDER EING
EBEN"
70 FOR N=1 TO D
80 PRINT"TEMPERATUR AM TAG";N
90 INPUT T(N)
100 NEXT N
110 LET SUM=0
120 LET MAX=T(1)
130 LET MIN=T(1)
140 FOR N=1 TO D
150 LET SUM=SUM+T(N)
160 IF T(N)>MAX THEN LET MAX=T(N)
170 IF T(N)<MIN THEN LET MIN=T(N)
180 NEXT N
190 PRINT
200 PRINT
210 PRINT"DURCHSCHNITTLICHE TEMPERATUR:"
;SUM/D
220 PRINT"MAXIMALE TEMPERATUR:";MAX
230 PRINT"MINIMALE TEMPERATUR:";MIN
```

Leerzeile

maximale Zahl an Monatstagen

Werte-Initialisierung

Beachte die EINGABESCHLEIFE in den Zeilen 70-100: ein zusätzlicher Weg, Daten in einem Feld abzulegen. Der Nachteil: Du mußt die Daten jedesmal eingeben, wenn Du das Programm startest. Die Alternative wäre das Speichern als DATEI auf einer Kassette oder Diskette, doch das geht über das Ziel dieses Buches hinaus.

Wenn Du jedoch keine Lust hast, einen Monat im Garten zu verbringen und ein Thermometer zu schütteln, bei Regen, Schnee und Wind, dann kannst Du auch die von Marta und Max Monsun gesammelten Daten verwenden:

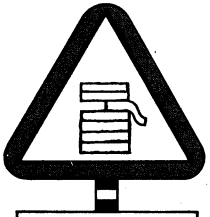
Temperaturtabelle

Beobachter: Marta und Max Monsun

Wetterstation: Kleinhinterwald

Monat: September 1984

Tag	Temperatur	Tag	Temperatur	Tag	Temperatur
1	11	11	15	21	10
2	13	12	14	22	11
3	14	13	13	23	16
4	9	14	10	24	17
5	21	15	9	25	16
6	16	16	13	26	20
7	15	17	11	27	19
8	15	18	8	28	20
9	3	19	2	29	14
10	17	20	7	30	16
				31	-



Abweichungen der Feldeinteilung

Wie hoch war die durchschnittliche, maximale und minimale Temperatur in dem verschlafenen Städtchen Kleinhinterwald im September 1984?

Regeln für Feldnamen

Die Regeln für die Bildung von Feldnamen sind die gleichen wie die für Variablennamen. Ver-

wende für NUMERISCHE Felder, also solche, die Zahlen speichern, Namen, die auch für numerische Variablen geeignet wären. (Es gibt auch STRING-FELDER; sie gehen aber über das Ziel dieses Buches hinaus.)

Du mußt ein Feld nicht dimensionieren, wenn es 10 oder weniger Einträge umfassen soll. Ein Feld S kann auch den Eintrag S(0) haben. Diese Dinge sind zwar nützlich, können aber auch leicht verwirren; ich rate Dir deshalb, sie wieder zu vergessen, bis Du alles andere in diesem Buch verstanden hast.

> Du kannst jeden korrekt gebildeten Namen für ein Feld verwenden - vergiß aber nicht die Klammern! NIMM5 ist eine ganz gewöhnliche Variable; NIMM(5) hingegen bezeichnet den Eintrag eines Feldes.

Mausschlaus Musik-Show

Margarete Mausschlau und ihr Mann Martin, der matte Mathematiker, haben mit dem Verkauf von Computerprogrammen ein hübsches Sümmchen verdient. Sie beschließen, das Geld in eine Theaterproduktion mit dem Namen

MAUSSCHLAUS MULTIMEDIALE MUSIK-SHOW

zu investieren. Sie besteht aus sechs Akten. Die Dauer jeder Vorstellung ist in dieser Übersicht verzeichnet:

1. Willi Wild und die Wiener Würstchen	11 Minuten
2. Mystro, der mäßige Magiker	8 Minuten
3. Das Holzwurm-Quartett	13 Minuten
4. Oh-La-La Can-Can-Chor	25 Minuten

5. Ludwig Läller, der österreichische Tenor 2 Minuten
6. Karl Koma, der chaotische Komiker 7 Minuten

Die Mausschlaus wollen die Zeiten in ihrem Computer speichern und die durchschnittliche Dauer eines Aktes errechnen. Wie würdest Du das machen?

Nehmen wir an, die Mausschlaus wollen wissen, welcher Akt der kürzeste ist. Welche Änderungen müssen an dem Programm vorgenommen werden?

Antworten

Übungsaufgaben

1. (a) 8+6=14
   (b) 5–9=–4
   (c) 2*1=2
   (d) 5*9=45
   (e) S(4)=2
   (f) S(6)*S(6)–2*S(5)+S(7)=1*1–2*0+9=10

2. Es gibt viele Lösungswege. Hier mein Vorschlag:

   0 S(5)
   1 S(2)–S(3)
   2 S(4)
   3 S(4)+S(6)
   4 S(2)–S(4)
   5 S(3)
   6 3*S(4)
   7 S(2)+1
   8 S(4)*(S(3)–S(6))
   9 S(7)

   Zur Probe fügst Du zum Beispiel folgende Programmzeile:

100 **PRINT** S(1)+S(2)

den Zeilen 10-80 (oder 10-60, Seite 70) zu, die das Feld aufbauen.

3.

```
10 DIM W(7)
20 LET W(1)=24
30 LET W(2)=93
40 LET W(3)=11
50 LET W(4)=14
60 LET W(5)=6
70 LET W(6)=2
80 LET W(7)=18
90 PRINT"TAG EINGEBEN (1-7)"
100 INPUT TAG
110 PRINT"DIE WINDGESCHWINDIGKEIT BETRUG
";
120 PRINTW(TAG)
```

Durchschnitt

Ändere Zeile 140 zu:

140 **PRINT**"DURCHSCHNITTLICHER SON NENSCHEIN PRO TAG IST";SUM/7; "STUNDEN."

Für schnelle Denker

Nein: MAX würde die ganze Zeit 10 bleiben. Du mußt einen kleinen Startwert wählen, so daß mindestens ein S(N) gleich oder größer ist. Ein sehr guter Startwert ist übrigens:

100 **LET** MAX=S(1)

Minimaler Sonnenschein

```
100 LET MIN=1000
    ↑
    für eine Minimum-Suche sehr geeignet
110 FOR N=1 TO 7
120 IF S(N)<MIN THEN LET MIN=S(N)
130 NEXT N
140 PRINT"MINIMALER SONNEN
    SCHEIN IST";MIN
```

Zeile 100 könnte auch ersetzt werden durch:

```
100 LET MIN=S(1)
```

Noch einige Aufgaben

Vielleicht mußtest Du bei diesen Aufgaben ganz schön nachdenken. Hier sind meine Lösungsvorschläge:

5.
```
100 FOR N=1 TO 7
110 IF S(N)<1 THEN GOTO 140
120 IF S(N)>5 THEN GOTO 140
130 PRINT N
140 NEXT N
```

Beachte, daß die Zeilen 110 und 120 den Computer veranlassen, die Tagesnummer N (in Zeile 130) NICHT auszugeben, wenn die Dauer des Sonnenscheins nicht im geforderten Bereich 1-5 liegt.

6.
```
100 FOR N=1 TO 7
110 IF S(N)=0 THEN PRINT"DIE SONNE
    HAT AM TAG";N;" NICHT
    GESCHIENEN."
120 NEXT N
```

7. 100 **LET** DU=(S(6)+S(7))/2
110 **PRINT**"AM WOCHENENDE SCHIEN DIE SONNE DURCH SCHNITTLICH";DU;" STUNDEN."

8. 100 **IF** S(7)>S(6) **THEN PRINT**"MEHR AM SONNTAG."
110 **IF** S(6)>S(7) **THEN PRINT**"MEHR AM SAMSTAG."
120 **IF** S(6)=S(7) **THEN PRINT**"AN BEI DEN TAGEN GLEICH LANG."

Zur Erinnerung! Versichere Dich, daß die Zeilen 10-80 (oder 10-60; Seite 69 oder 70) im Speicher des Computers stehen, bevor Du eines der obigen Programme startest; außerdem solltest Du prüfen, ob sich außer den gewünschten Zeilen nicht noch andere im Computer befinden.

Das geht am besten, indem Du die unerwünschten Zeilen löschst. Wenn Du **NEW** eingibst, mußt Du jedesmal die Zeilen wieder eingeben, die das Feld aufbauen. Das wird nach dem dritten oder vierten Mal langweilig.

Computer-Experiment

DURCHSCHNITTLICHE TEMPERATUR: 13
MAXIMALE TEMPERATUR: 21
MINIMALE TEMPERATUR: 2

Mausschlaus Musik-Show

```
10 DIM D(6)
20 LET D(1)=11
30 LET D(2)=8
40 LET D(3)=13
```

```
50 LET D(4)=25
60 LET D(5)=2
70 LET D(6)=7
100 LET SUM=0
110 FOR N=1 TO 6
120 LET SUM=SUM+D(N)
130 NEXT N
140 LET DU=SUM/6
150 PRINT"DIE DURCHSCHNITTLICHE DAUER IS
T:";DU
```

Um den kürzesten Akt zu ermitteln, änderst Du die Zeilen 100-150 wie folgt:

```
10 DIM D(6)
20 LET D(1)=11
30 LET D(2)=8
40 LET D(3)=13
50 LET D(4)=25
60 LET D(5)=2
70 LET D(6)=7
100 LET MIN=D(1)
110 FOR N=1 TO 6
120 IF D(N)<MIN THEN LET MIN=D(N)
130 NEXT N
140 FOR N=1 TO 6
150 IF D(N)=MIN THEN PRINT"DER KUERZESTE
 AKT IST NUMMER";N
160 NEXT N
```

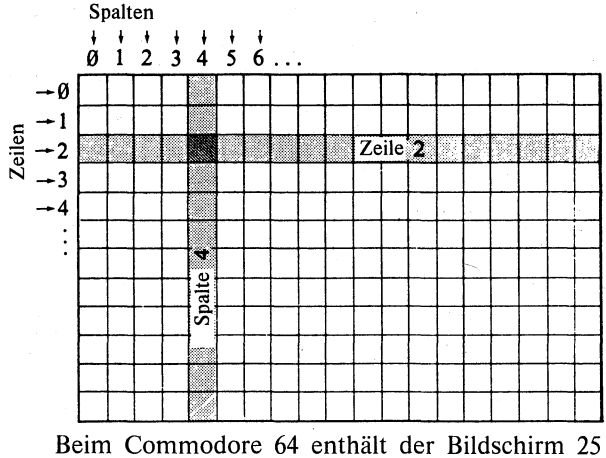
7

Squire Stoatthrostle und das TAB

Bis jetzt haben wir der Frage, *WO* Ausgaben auf dem Bildschirm erscheinen, noch keine große Aufmerksamkeit geschenkt; Hauptsache, die Daten erschienen dann, wenn wir sie benötigten. Die Vernachlässigung dieser Frage war aber unseren Programmierkünsten angemessen: Wenn man noch dabei ist, das Programmieren zu lernen, will man vorerst nicht mit Details belästigt werden.

Doch nun ist der Zeitpunkt gekommen (sagte das Walroß ...), sich um eine Möglichkeit der Bildschirmordnung zu kümmern: die Tabulation.

Deine Bildschirmanzeige ist in eine Reihe von Zellen eingeteilt, die jeweils ein Zeichen (Buchstabe, Ziffer, Grafiksymbol usw.) enthalten. Die waagrechten Linien aus diesen Zellen sind die Bildschirmzeilen; die senkrechten sind die Bildschirmspalten.



Beim Commodore 64 enthält der Bildschirm 25 Zeilen und 40 Spalten. In diesem Kapitel interessieren uns die Spalten.

S
P
A
Z E I L E N liegen waagrecht
T
E
N stehen senkrecht

Die Lummoxshire-Liga

Wir machen nun einen Ausflug in die englische Provinz.

Etwa zehn Meilen entfernt von der Ortschaft Flannel-under-Ware liegt das Dörfchen Hogwallow. Obwohl es nur eine 122-Seelen-Gemeinde ist, hat es eine der besten Fußballmannschaften der Umgebung: die Hogwallow Hackers. Jedes Jahr spielt diese Mannschaft in der dortigen nach dem Landkreis benannten Lummoxshire-Liga gegen so bekannte Teams wie Bumbleforth Benighted, Prongsworthy Ravers, Cow Green Agricultural and Mechanical, Womblehampton Waverers und gegen die Gentle Elastic Company, eine ansässige Kunstgummifabrik.

Im Moment sieht die Punktetabelle der Liga so aus:

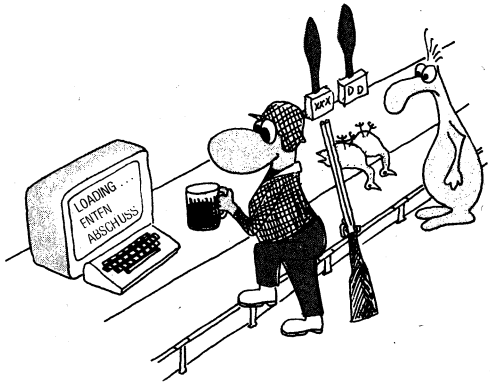
Offizielle Punktetabelle der Lummoxshire-Liga

Hogwallow Hackers

Gegner	Tore	Gegentore	Punkte gesamt
Bumbleforth	7	0	2
Prongsworthy	2	2	3
Cow Green	4	3	5
Womblehampton	4	5	5
GEC	3	1	7

Punkte: Siege 2, Unentschieden 1, Niederlagen 0.

Es ist 20.30 Uhr. Wir befinden uns in der Schankstube des Wirtshauses „Zum rheumatischen Wildschwein", dem einzigen Lokal von Hogwallow. Die Hackers haben soeben die Twangers (Näseler, Spitzname der Mannschaft der Gentle Elastic Company) geschlagen; es geht also entsprechend hoch her. Der Kapitän und der Manager der Hackers, Alf Thyme und Jock Strappe, sind gerade dabei, die letzten Spielergebnisse auf einer speziellen, neben der Kegel-Tabelle angebrachten Tafel zusammenzufassen ...



Alf: Mann, das war'n gutes Spiel!

Jock: Klar, Alf. Wo is'n die verdammte Kreide hin? He, Mary! Hat wer die Kreide geklaut?

Mary (die Kellnerin): Woher soll isch das wiss'n? Bob Frapples hat se letzte Nacht beim Darts (das typisch britische Pfeilwurfspiel) gegen die Bumbleforth B-Mannschaft gebraucht.

Jock: Das gibt's doch nich', Mary! Hat der Kerl schon das fünfte Mal die Kreide gebunkert! Wie soll isch dann jetzt die Pungde uffschreibe?

Alf: Weiß'te, Jock, vielleischt sollden wir unsere Methoden modernisieren. Isch hab' in der „Ziegenzüschter-Zeitung" 'was geles'n, wie eins von diese' neue' Combjuder-Dingern die Zieschezucht

total revolutioniere tut. Un' da is mir eing'falle, daß die Hackers ja auch auf so'me Ding ihre Pungde notiere könnde.

Jock: Was'n Schwachsinn, Alf! 'Ne Pungdetabelle auf 'ne Ziege schreibe, also wiklisch!

Alf: Nein! Net auf 'ne Ziege, auf'n Combjuder!

Jock: Das wär was! Oder, Marry? Ho, ho, ho!

Alf: Nein, Jock, isch mein' des enst! Guck mal, da is doch dieser Squire Stoatthrostle. Isch hab' gehört, daß der vor paar Wochen einen gekauft hat. (Er wendet sich zu Squire.) He, Squire!

Squire S.: Guten Abend! Danke, Mary, das übliche. Doppelten Wodka und einen Antifrost.

Alf: Squire, isch hab' so was läuten hören, als ob du dir eins von diese Combjuder-Teufelsdingern geholt hättest.

Squire S.: Richtig, einen Stork-37 mit zwei Diskettenlaufwerken und 64 Kilobyte RAM.

Alf: Siehst'e, Jock, der Squire is schon ein rischtiger Experde! Weißt'e, Squire, wir haben uns nämlisch gerade überlegt, ob man mit einem Combjuder net unsere Pungdetabelle verwalde könnt. Dann könnde mir des ganse auf'm Fernseher uns agugge.

Mary: Aber net, wenn die Straße von San Fransisco laufe! Dann gibt's Äscher!

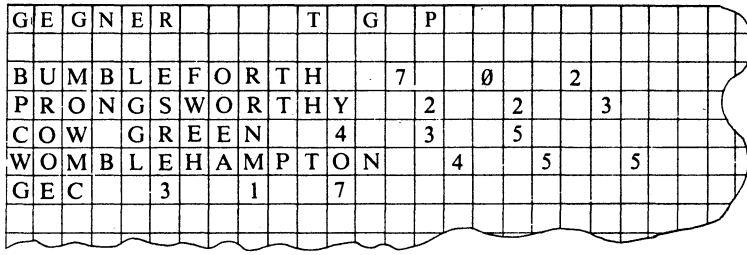
Jock: Das möscht isch sehn! Beim letzte' Mal . . .

***

Und das Gespräch schweifte ab. Doch noch am selben Abend beschloß Squire Stoatthrostle, daß es doch ganz lustig sein könnte, die Anregung aufzunehmen. Nachdem er einen längeren Blick auf Kapitel 3 geworfen hatte, entschied er sich für die Verwendung von DATA-Befehlen. Er gelangte schließlich zu folgendem Programm:

```
10 DATA BUMBLEFORTH,7,0,2
20 DATA PRONGSWORTHY,2,2,3
30 DATA COW GREEN,4,3,5
40 DATA WOMBLEHAMPTON,4,5,5
50 DATA GEC,3,1,7
60 PRINT"GEGNER T G P"
70 PRINT
100 FORN=1TO5
110 READG$,T,G,P
120 PRINTG$;" ";T;G;P
130 NEXTN
```

Als Alf und Jock am nächsten Abend ins „Rheumatische Wildschwein" schlenderten, waren sie nicht wenig überrascht, dies auf dem Bildschirm zu sehen:



Squire war sehr stolz auf sein Programm, aber obschon Alf und Jock es sehr höflich bewunderten, merkte er doch, daß sie nicht ganz zufrieden waren.

Warum?

Stoatthrostles Schnitzer

Endlich bemerkte Squire, was die beiden störte. Wie es Noddy Numskull, der Dorfdepp, formu-

lierte, sahen die Spalten aus „wie die Hinterbeine eines gichtkranken Hammels". Die Ergebnistabelle war unordentlich und schwer zu lesen, ungeeignet dazu, mit einem kurzen Blick die gewünschte Information zu entnehmen.

Stoatthrostle wußte, daß etwas geschehen mußte.

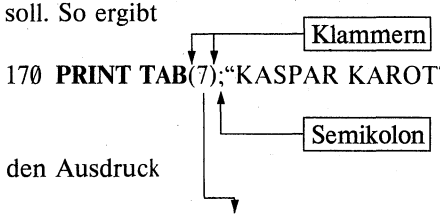
Aber was?

TAB

Die Antwort ist das BASIC-Befehlswort

TAB

Es teilt dem Computer mit, daß er beim Ausdrukken in einer gegebenen Bildschirmspalte beginnen soll. So ergibt



den Ausdruck

```
Spalte 0 1 2 3 4 5 6 7 8 9 . . .
                    KASPAR KAROTTE
```

Allgemein druckt ein Befehl wie

120 **PRINT** TAB(Zahl);String

den angegebenen String in der aktuellen Bildschirmzeile ab der durch die Zahl bezeichneten Spalte aus. Wenn die angegebene Spalte in dieser Zeile schon überschritten wurde, wird der angegebene String ab der nächsten freien Druckposition der Zeile ausgegeben. Nun betrachte

(a) 1Ø **PRINT TAB**(2);"ZWEI";**TAB**(11);"ELF"
(b) 1Ø **PRINT TAB**(11);"ELF";**TAB**(2);"ZWEI"

In (b) wird der String "ZWEI" direkt hinter "ELF" gedruckt, denn "ELF" wurde schon auf eine Spalte gesetzt, die hinter Spalte 2 liegt. (Bei manchen Computern würde "ZWEI" in Spalte 2 der folgenden Zeile landen.)

Um die Ausgabe ordentlicher zu gestalten, verwendet man

TAB

in **PRINT**-Befehlen. Vergiß nicht die Klammern.

Stoatthrostle denkt noch einmal nach

Nachdem Squire Stoatthrostle einige Winke mit Zaunpfählen nichts genutzt hatten, schenkten die Hogwallow Hackers ihm schließlich das Buch „Das Tor zur Computerwelt, Band 2". Dort fand er

auf Seite 88 das Kapitel, das sein Problem behandelte. (Hallo, Squire! Wie geht's denn so?) Nachdem er den Abschnitt **TAB** gelesen hatte, war er in der Lage, einige entscheidende Verbesserungen an seinem Programm vorzunehmen. Ich möchte Squire hiermit nochmals für die Lösung danken, die im folgenden abgedruckt ist. Mich beschleicht allerdings der vage Verdacht, daß er sie von jenem Buch abgeschrieben hat, das ihm die Hackers geschenkt hatten.

Wie dem auch sei, hier ist sie:

```
10 DATA BUMBLEFORTH,7,0,2
20 DATA PRONGSWORTHY,2,2,3
30 DATA COW GREEN,4,3,5
40 DATA WOMBLEHAMPTON,4,5,5
50 DATA GEC,3,1,7
60 PRINT"GEGNER";TAB(20);"T";TAB(23);"G"
;TAB(26);"P"
70 PRINT
100 FORN=1TO5
110 READG$,T,G,P
120 PRINTG$;TAB(20);T;TAB(23);G;TAB(26);
P
130 NEXTN
```

Das Prinzip ist einfach: Die Spalten 0, 20, 23, 26 werden als Standardpositionen für die unter „Gegner", „Tore", „Gegentore" und „Punkte gesamt" eingetragenen Werte benutzt. Dabei stellt man fest, daß man den Befehl TAB(0) zu Beginn einer PRINT-Zeile wie in

PRINT TAB(0);"GEGNER"; usw.

nicht benötigt, denn der Computer beginnt ohnehin mit jeder Zeile in Spalte 0. Hier der neue Ausdruck:

Ø																		20			23			26
G	E	G	N	E	R													T			G			P
B	U	M	B	L	E	F	O	R	T	H								7			Ø			2
P	R	O	N	G	S	W	O	R	T	H	Y							2			2			3
C	O	W		G	R	E	E	N										4			3			5
W	O	M	B	L	E	H	A	M	P	T	O	N						4			5			5
G	E	C																3			1			7

Jetzt ist alles in Ordnung. Das „Rheumatische Wildschwein" mußte anbauen, um die Festbesucher fassen zu können.

Seufzerspalte

Falsche Zahlen

Wenn Du soeben auf die Wahrheit meiner Worte vertraut hast und Squires Programm nicht auf Deinem Computer ausprobiert hast, dann möchte ich Dir raten, dies so schnell wie möglich nachzuholen.

Wie Du siehst, habe ich gelogen.

Es funktioniert nicht ganz!

Die Zahlen in den T-G-P-Spalten werden alle eine Spalte zu weit rechts ausgegeben. Ich werde gleich erklären, warum; aber zuerst möchte ich Dir ein wichtiges Fehlerbeseitigungsprinzip vorführen:

> Um einen Fehler verbessern zu können, mußt Du nicht immer verstehen, *warum* er aufgetreten ist.

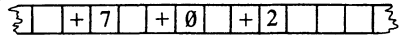
Um diesen Satz zu beweisen, wollen wir einfach einmal versuchen, in Zeile 120 alles eine Spalte nach links zu bewegen:

```
120 PRINT G$;TAB(19);T;TAB(22);G;TAB(25);P
                  ↑          ↑          ↑
          alle TABs werden um 1 erniedrigt
```

Laß es noch einmal laufen. Potztausend, es geht! Unser Lösungsweg ist zwar etwas eigenartig, aber es geht!

Ein zu Null für uns. Es ist aber natürlich im übrigen und überhaupt viel befriedigender zu wissen, *warum* das Zeug an der falschen Stelle erschien.

Die BASIC-Version Deines Computers ist an einem Industriestandard orientiert, dem Microsoft-BASIC. Im Microsoft-BASIC werden alle Strings dort ausgedruckt, wo man es erwartet, Zahlen allerdings manchmal nicht. Microsoft hat nämlich die Vorgabe, daß jede Zahl, die nicht negativ ist, ein „unsichtbares" Plus-Zeichen (+) vor sich hat. (Zahlen, die negativ sind, haben ein weithin sichtbares Minus-Zeichen!) Der Rechner setzt also in die Spalten 20, 23 und 26 statt der Zahlen 7, 0 und 2 deren „+"-Zeichen:



Die „+"-Zeichen kann man nicht tatsächlich sehen (sie sind wie gesagt unsichtbar), aber sie besetzen immer noch den Raum in der betreffenden Spalte. Das verschiebt die eigentlichen Zahlen nach rechts. (Natürlich druckt die Maschine ebenfalls ein LEERZEICHEN *hinter* der Zahl - hier also in den Spalten 22, 25, 28 -, dieses wird aber nicht durch den **TAB**-Befehl hervorgerufen; numerischen Variablen wird *immer* noch ein LEERZEICHEN angehängt.)

Eine negative Zahl, wie –7, wird auf die gleiche Weise ausgegeben, nur mit dem Unterschied, daß

diesmal das Vorzeichen, das „–"-Zeichen, sichtbar ist und somit alles korrekt aussieht. Der Gedankengang, auf dem diese etwas bizarre Vereinbarung beruht, ist der, daß Zahlenspalten sauberer aussehen, wenn für das Vorzeichen ein Platz reserviert wird. Andererseits würde kein vernünftiger Mensch das Plus-Zeichen tatsächlich hinschreiben, also ...

Versuche dieses Beispielprogramm:

```
10 LET X = 1
20 FOR T = 1 TO 10
30 LET X = -1.5 * X
40 PRINT TAB(5); X
50 NEXT T
```

Du erhältst einen Ausdruck mit abwechselnd positiven und negativen Zahlen, etwa diesen

```
 - 1 . 5
   2 . 2 5
 - 3 . 3 7 5
   5 . Ø 6 2 5
 - 7 . 5 9 3 7 5
   1 1 . 3 9 Ø 6 2 5
```

usw. Beachte, daß trotz der heldenhaften Bemühungen von seiten der Firma Microsoft der Dezimalpunkt in der sechsten Zeile aus der Reihe tanzt: Man kann es nicht allen recht machen.

> Wenn durch **TAB** eine Zahl eine Spalte zu weit rechts erscheint, dann ist der Grund meistens ein „unsichtbares Plus-Zeichen". Verringere die **TAB**-Zahl um eins.

Martins Exponential-Übung

Martin Mausschlau, der multidimensionale Mathematiker, will auf dem Bildschirm

Quadrate	N*N
Kuben	N*N*N
Vierte Potenzen	N*N*N*N

zu den Zahlen von 1 bis 20 ausgeben lassen. Die Zahlen sollen folgendermaßen erscheinen:

Spalte	6	12	18	24
	↓	↓	↓	↓
	N	N*N	N*N*N	N*N*N*N
	1	1	1	1
	2	4	8	16
	3	9	27	81

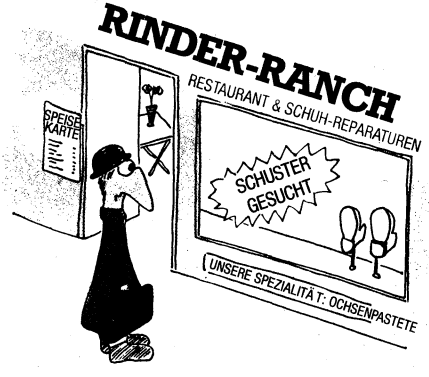
Wie kann er dies mit Hilfe von **TAB** bewerkstelligen?

Der Verzweifelte Volker

Der Verzweifelte Volker, Du erinnerst Dich vielleicht an ihn, liebt Ochsenpastete. Er arbeitet gerade an einem Entwurf einer Fernsehreklame für eine Restaurantkette, die als „Rinder-Ranch, Restaurant und Schnellschuhmacher" bekannt ist und von hohen Regierungsbeamten unterstützt wird. Bei dieser Reklame soll das Wort „OCHSENPASTETE" diagonal auf dem Bildschirm erscheinen. Etwa so:

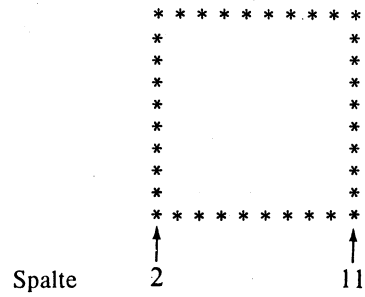
```
O
 C
  H
   S
    E
     N
      P
       A
        S
         T
          E
           T
            E
```

Wie kann man das erreichen?



Roberts Rahmen

Robert experimentiert mit Computer-Graphiken. Er hat eine ganz besonders originelle Idee: Er will einen Rahmen aus Asterisken (*) mit 10 Zeichen Kantenlänge ausdrucken:



Kannst Du Dir denken, wie er das am besten macht?

(Tip: Benutze den Befehl PRINT"**********" für Ober- und Unterkante und ein paar TABs für die Seiten.)

Antworten

Martins Exponential-Übung

```
10 PRINTTAB(6);"N";TAB(12);"N*N";TAB(18)
;"N*N*N";TAD(24);"N*N*N*N"
20 FORN=1TO20
30 PRINTTAB(5);N;TAB(11);N*N;TAB(17);N*N
*N;TAB(23);N*N*N*N
40 NEXTN
```

Der Verzweifelte Volker

Dieses Problem schreit nach brutaler Gewalt und Stupidität. (Es sind anspruchsvollere Lösungen möglich, zu diesen fehlen uns jedoch einige Kenntnisse.)

```
10 PRINT"O"
20 PRINTTAB(1);"C"
30 PRINTTAB(2);"H"
40 PRINTTAB(3);"S"
50 PRINTTAB(4);"E"
60 PRINTTAB(5);"N"
70 PRINTTAB(6);"P"
80 PRINTTAB(7);"A"
90 PRINTTAB(8);"S"
100 PRINTTAB(9);"T"
110 PRINTTAB(10);"E"
120 PRINTTAB(11);"T"
130 PRINTTAB(12);"E"
```

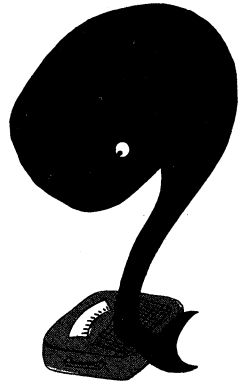
Du kannst es auch mit einer Schleife und DATA-Befehlen probieren.

Roberts Rahmen

```
10 PRINTTAB(2);"**********"
20 FORK=1TO8
30 PRINTTAB(2);"*";TAB(11);"*"
40 NEXTK
50 PRINTTAB(2);"**********"
```

8
Logik

1. Nur Elefanten oder Wale bringen Junge zur Welt, die über 100 Kilogramm wiegen.
2. Der Sohn des Bundeskanzlers wiegt 110 Kilogramm.
3. Also ist . . .



LOGIK ist die Lehre, aus wahren Behauptungen wahre Schlußfolgerungen zu ziehen. Die obenstehende Schlußfolgerung amüsiert uns, da wir wissen, daß sie falsch ist. Dennoch erscheinen die einzelnen Schritte logisch. Der Trugschluß dabei ist, daß nur das Gewicht zum Zeitpunkt der Geburt zählt. Der Sohn des Bundeskanzlers war ein normales gesundes Baby von etwa fünf kg.

Die Logik befaßt sich mit *Behauptungen.* Eine Behauptung ist etwas, daß entweder *wahr* oder *falsch* ist (obwohl es manchmal sehr schwer sein kann zu bestimmen, was von beiden).

Beispiele für Behauptungen sind:

- 2+2=5 (falsch)
- Dieser Satz steht auf Seite 97. (falsch)
- Alle Kühe sind Säugetiere. (wahr)
- Alle Säugetiere sind Kühe. (falsch)
- Die Armee-Seriennummer von Elvis Presley war: 53310761. (wahr)
- Das Pferd Alexanders des Großen hieß Buzephalus. (wahr)
- Alexander der Große hatte ein Schaf namens William. (Wahrscheinlich falsch; aufgrund der fehlenden historischen Dokumente ist es aber nicht einfach zu entscheiden.)

Du kannst Dir keine Sätze vorstellen, die keine Behauptungen sind? Wie wär's damit?

- Häßliche, grüne, verrückte Dinge.
- Warum ist eine Maus, wenn sie sich dreht?
- Was ist der Unterschied zwischen einer Ente?
- Guten Morgen!
- Und so weiter.

Entweder ist es
WAHR
oder es ist
FALSCH
oder es ist keine Behauptung.

Aufgaben

Welche der folgenden Sätze sind Behauptungen? Welche von diesen Behauptungen sind wahr, welche sind falsch?

(a) Wie steht's, gescheckte Kuh?
(b) Giraffen haben lange Hälse.

(c) Was fährt mit 100 km/h die Wäscheleine rauf und runter?
(d) 14+22>5.
(e) 4. Juli 1776.
(f) Wenn Schweine Flügel hätten, würden wir alle Regenschirme tragen.
(g) Entweder schneit es oder nicht.
(h) Heute ist Dienstag oder 2+2=4.
(i) Sie sind der schwarze Mann und ich fordere meine 5 Mark.
(j) Wenn heute Dienstag ist, dann ist morgen Mittwoch.
(k) Zweimal Spaghetti Bolognese.
(l) Dieser Satz ist falsch.

Verknüpfte Behauptungen

Manchmal werden Behauptungen aus anderen Behauptungen zusammengesetzt, indem man diese mit den Wörtern

UND
ODER

verknüpft. Zum Beispiel:

(m) Es ist Montag **UND** mir ist langweilig, langweilig, langweilig, langweilig, langweilig.
(n) Rosen sind rot **UND** Veilchen sind violett.
(o) Ich komme mit dem Auto **ODER** mit dem Zug.
(p) Diese Schuhe sind zu klein **ODER** meine Füße sind zu groß.

Schau Dir in diesem Zusammenhang auch noch einmal (g, h, i) an. Diese Behauptungen nennt man VERKNÜPFTE Behauptungen. Bei Computern werden sie in Verbindung mit dem IF ...

THEN-Befehl verwendet, um bestimmte Anweisungen auszuführen, wenn eine Reihe von Behauptungen wahr ist. Der Computer muß aber noch einige Seiten warten. Zuerst müssen wir feststellen, wann solche verknüpften Behauptungen wahr sind und wann nicht.

Nehmen wir als Beispiel die Behauptung (m) von oben. Wann ist sie wahr?

Nehmen wir an, heute ist Donnerstag. Dann ist (m) offensichtlich FALSCH. Auch wenn mir wirklich langweilig, langweilig, langweilig, langweilig, langweilig ist.

Nehmen wir an, es ist tatsächlich Montag, aber anstatt gelangweilt zu sein, bin ich bester Laune, voller Tatendrang, strahle über beide Ohren und will gerade gehen. Dann ist (m) immer noch falsch.

Demnach müssen in einer mit **UND** verknüpften Behauptung beide Teile wahr sein, wenn die gesamte Behauptung wahr sein soll. Wir können in einer kleinen Tabelle alle Möglichkeiten darstellen:

Es ist Montag	Mir ist langweilig, langweilig, langweilig, langweilig, langweilig	Es ist Montag **UND** mir ist langweilig, langweilig, langweilig, langweilig, langweilig
wahr	wahr	wahr
wahr	falsch	falsch
falsch	wahr	falsch
falsch	falsch	falsch

Wie steht's mit **ODER**? Betrachten wir uns zu diesem Zweck Beispiel (p). Nehmen wir an, diese Schuhe sind NICHT zu klein, meine Füße sind aber nichtsdestotrotz zu groß. Dann kann man mit Fug und Recht sagen, daß (p) wahr ist. (Es wird nicht

behauptet, daß beide Aussagen zutreffen, nur die eine **ODER** die andere.) Die Sache funktioniert genauso, wenn meine Füße perfekt geformt und alles andere als zu groß, die Schuhe aber deformiert und unbequem sind, weil sie irgendein Idiot in der Fabrik fünf Nummern kleiner geschustert hat, als es auf dem Karton steht. Jetzt ist die verknüpfte Behauptung wahr, wenn eine (möglicherweise auch beide) Teilbehauptungen zutreffen. Um falsch zu sein, müssen beide Teile falsch sein. In Tabellenform sieht das so aus:

Diese Schuhe sind zu klein	Meine Füße sind zu groß	Diese Schuhe sind zu klein **ODER** meine Füße sind zu groß
wahr	wahr	wahr
wahr	falsch	wahr
falsch	wahr	wahr
falsch	falsch	falsch

Ein Logiker würde das alles kurz zusammenfassen, indem er die beiden Teilbehauptungen P und Q nennen und W und F für wahr und falsch setzen würde:

P	Q	P **UND** Q	P **ODER** Q
W	W	W	W
W	F	F	W
F	W	F	W
F	F	F	F

Eine solche Tabelle nennt man *Wahrheitstafel.*

Sie komprimiert das Problem auf kleinstem Raum; aber eigentlich ist die Sache sehr simpel. Logiker sind überhaupt sehr simple Leute.

Du kannst

UND

und/oder

ODER

verwenden, um mehrere Behauptungen zu einer zu verknüpfen.

Die Kartoffeln der Gräfin

Es läutete an der Tür und ich öffnete, um zu sehen, wer es sei. Es war ein Postbote auf einem Fahrrad. Ich nahm den schlanken, braunen Briefumschlag entgegen und gab dem Mann für seine Mühe einen Schilling.

„Holmes, Holmes, ein Telegramm!"

Der große Mann war gerade in einen komplizierten Versuch mit einer Blutprobe und einigen Chemikalien vertieft. Er schüttelte ein Reagenzglas, das mit einer blauen Flüssigkeit gefüllt war. „Watson, wenn diese Flüssigkeit gelb wird, bedeutet das das Leben eines Mannes! Ein hübsches Beispiel der Rechtskunst, wenn ich so sagen darf, sehen Sie!"

„Aber Holmes, das ist grün, nicht gelb!"

Telegramm

Königliche Schneckenpost
Sicher, zuverlässig, langsam

AN MR. SHERLOCK HOLMES, 221B BAKER ST., LONDON.
VON LADY INDORA BADEN-POOLE, GRÄFIN VON WESTHAMPTONSHIRE

SCHWEINE HABEN FLÜGEL, UND KÖNIGIN ANNE IST TOT; ODER DER FROSCH KERMIT IST PRÄSIDENT DER USA; ODER DIE PREISGEKRÖNTE KARTOFFEL DES GRAFEN WURDE ENTFÜHRT; ODER SPINNEN HABEN NEUN BEINE; ODER 'CHACUN A SON GOUT'IST FRANZÖSISCH UND BEDEUTET 'JEDER HAT NOCH EINEN GUT'. KOMMEN SIE SOFORT, HONORAR WIE GEWÖHNLICH, INDORA.

„Nah genug, Watson. Ich bin sowieso farbenblind. Haben Sie das Telegramm geöffnet?"

„Ja, Holmes. Aber - Ich verstehe nicht ein einziges Wort daraus!"

Er nahm es mir aus der Hand.

„Mit Sicherheit, Watson", sagte Holmes gereizt, „können Sie das Wort SCHWEINE verstehen?"

„Das natürlich, Holmes -."

„Und eben haben Sie noch behauptet, Sie würden nicht ein einziges Wort verstehen."

„Verdammt, Holmes! Was ich sagen wollte, war: Ich verstehe den Sinn nicht! Warum müssen Sie nur immer so kaltblütig und logisch sein?"

„Aber, mein Gott, Watson, der Sinn ist doch völlig klar; und eine logische Analyse ist der Schlüssel zu seinem Verständnis. Die einzige Frage, die wir noch beantworten müssen, ist: IST ES WAHR?"

Ich runzelte die Stirn. Intelligente Einfälle sind mir noch nie von selbst gekommen. „Nun ... angenommen, es WÄRE wahr, Holmes - was würde es dann BEDEUTEN?"

„Das", sagte Sherlock Holmes, „müssen Sie herausfinden".

Kannst Du Watson aus seiner mißlichen Lage helfen? Angenommen, die GANZE BEHAUPTUNG ist WAHR, welche Nachricht hat das Telegramm zum Inhalt? (Ein Tip für den deutschen Leser: Königin Anne ist tot und Kermit ist nicht ...)

Die traurige Geschichte von Stefan Schieber

Stefan Schieber, der skurrile Software-Spezialist, versucht dem Gasmagnaten J. Paul Igitty ein Aktienmarkt-Programm zu verkaufen, das von der Firma Softbirne für den Stork-37 Mikro entwickelt wurde. Es besteht aus zwei Teilen. Der erste Teil stellt die

Kurse von Aktien und Wertpapieren auf dem Bildschirm dar. Der zweite erlaubt es dem Benutzer zu wählen, welche Papiere er kaufen oder verkaufen will. Das Programm nimmt dann automatisch über das Mikronit-Netzwerk mit dem Computer des Börsenmaklers Kontakt auf, um die Transaktion durchzuführen.

Aus urheberrechtlichen Gründen sind wir nicht in der Lage, das ganze Programm hier abzudrucken; die wichtigsten Segmente findest Du jedoch weiter unten. Wir wollen an dieser Stelle nicht vergessen, der Firma Softbirne für ihre freundliche Unterstützung zu danken.

```
10 PRINT"TEIL EINS"
20 PRINT"ANALYSE DES AKTIENMARKTES"

   (die folgenden 1970 Zeilen wurden gelöscht)

2000 PRINT"WOLLEN SIE WEITERMACHEN?"
2010 PRINT"GEBEN SIE IHRE ANTWORT EIN:
     J/N"
2020 INPUT A$
2030 IF A$="NEIN" THEN GOTO 7000
2040 PRINT"TEIL ZWEI"
2050 PRINT"AUTOMATISCHE AKTIEN-TRANS
     AKTION"

     (die folgenden 4940 Zeilen wurden gelöscht)

7000 PRINT"VIELEN DANK FÜR DIE
     VERWENDUNG VON SOFTBIRNE-
     PROGRAMMEN."
7010 PRINT"AUF WIEDERSEHEN."
7020 STOP
```

J. Paul Igitty war an diesem Programm außerordentlich interessiert. Er spielte sogar mit dem Gedanken,

600 Kopien für sein internationales Firmenkonsortium zu kaufen. Das wäre ein sehr lukrativer Auftrag für Stefan Schieber.

Das Programm arbeitete einwandfrei, bis Igitty die Zeile 2000 erreichte - ein Hinweis auf eine Eingabe, die es ihm erlaubt, das Programm zu verlassen, wenn er nicht mit dem zweiten Teil weitermachen will. Nun, mit der Antwort „Ja" auf jede dämliche Frage wird man kein Gasmagnat; und es war Igittys Prinzip, solange „Nein" zu sagen, bis er Zeit hatte, sich die Sache zu überlegen. Er bemerkte, daß Zeile 2010 J/N als Eingabe fordert und schloß daraus, daß „J" für „Ja" und „N" für „Nein" steht. (Erstens war er nicht von gestern, und zweitens wäre er nie auf die Idee gekommen, „J/N" einzugeben.)

Also tippte Igitty „N" und drückte RETURN

Aber anstatt das Programm zu verlassen, mußte er verwundert lesen:

TEIL ZWEI
AUTOMATISCHE AKTIEN-TRANSAKTION

Er brauchte weitere 20 Minuten, um das Programm endgültig zu verlassen (was ihn ungefähr 42074 Dollar an Zeitverschwendung kostete).

Vergebens wies ihn Stefan Schieber darauf hin, daß die korrekte Eingabe „NEIN" lautete und nicht „N". Das Programm hatte in Zeile 2030 mit "NEIN" verglichen und war, da das Ergebnis des Vergleichs „falsch" gewesen war, demzufolge nicht zu Zeile 7000 gesprungen.

J. Paul Igitty ließ sich nicht beeindrucken.

Stefan Schieber ging in sein Büro, rief die Leute in der Programmentwicklungsabteilung an und ließ sie genau wissen, was er von ihnen dachte.

Protokoll-Peter, der prosaische Programmierer, der gerade bei Softbirne angefangen hatte, schrieb das Programm um. Er änderte nur eine Zeile, nämlich:

Ferner machte er noch einige sarkastische Bemerkungen über die trivialen Aufgaben, mit denen man ihn betraute: Das wäre eine Verschwendung seiner bemerkenswerten Talente.

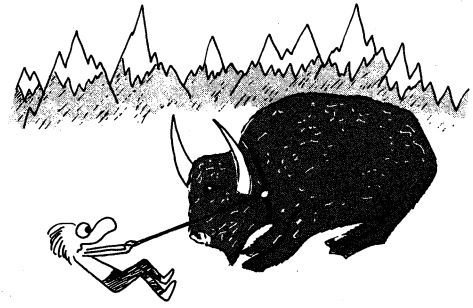
Stefan Schieber verbrachte zwei Stunden am Telefon, um die Sekretärin J. Paul Igittys zu überreden, ihm eine zweite Chance zu geben. Eine Woche später besuchte er Igitty in seinem Büro und startete einen neuen Versuch.

Alles ging glatt, bis Igitty zu Zeile 2010 kam. Er erinnerte sich, daß Programmierer oft „J/N" als Abkürzung für „Ja" oder „Nein" verwenden, und tippte deshalb:

NEIN

Nun . . . diesmal verglich der Computer die Eingabe mit „N" und sprang wieder nicht zum Programmende, da der Vergleich erneut negativ ausfiel. Noch einmal 42074 Dollar an Zeitverschwendung.

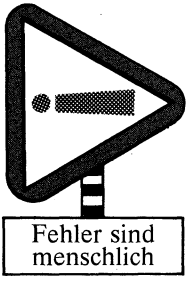
Stefan verlor den Auftrag und arbeitet jetzt als Yak-Herdentreiber in der Mongolei. Was wäre er jetzt, wenn er schon damals dieses Kapitel gelesen hätte?



Das bedienerfreundliche INPUT

Und jetzt kommen wir zu dem Teil, der Dir die Frage beantworten soll, die Dir schon seit einiger Zeit auf der Zunge brennt: WAS UM ALLES IN DER WELT HAT DAS MIT LOGIK ZU TUN?

Ein weitverbreitetes Problem: Computerbenutzer können oft wählen, wie sie im Programm fortfahren wollen. Doch bei der Eingabe ihrer Wahl kommt es manchmal zu Unstimmigkeiten: Schon die Verwechslungsgefahr zwischen „J" und „JA" oder zwischen „N" und „NEIN" ist groß.



Natürlich könntest Du lang und breit erklären, welche Eingaben zulässig sind, aber Leute, die keine Computer sind, machen Fehler. Es wäre doch viel einfacher, die Eingabe so anzulegen, daß es dem Benutzer überlassen bleibt, ob er „N" oder „NEIN" wählt.

Mit anderen Worten, wir wollen zu Zeile 7000 springen, wenn eine der beiden Bedingungen:

1. A$="NEIN"
2. A$="N"

erfüllt ist.

Und das erreichen wir mit

```
2030 IF A$ = "N" OR A$ = "NEIN" THEN GOTO
     7000
```

Bedingung 1 | Bedingung 2

(Das ODER aus der Logik wird in BASIC durch das englische OR ersetzt).

Stell Dir vor, wie froh J. Paul Igitty gewesen wäre! (Und Stefan Schieber erst.)

Programme, die so geschrieben sind, daß sie auch von ungeübten Personen bedient werden können, werden *benutzerfreundlich* genannt. Ein benutzer-

freundliches Programm zu schreiben, ist nicht nur eine Frage der *Programmiertechnik,* es ist eine Frage des *Stils.*

Vielleicht funktioniert ein Programm gut. Wenn es aber auf Grund verzeihbarer Fehler von seiten des Benutzers immer wieder das Falsche macht, dann ist es nicht benutzerfreundlich, es ist benutzerfrustrierend.

> Wird eine Eingabe auf Fehler untersucht, oder stehen mehrere inhaltlich gleiche Antwortmöglichkeiten zur Wahl, dann ist diese Eingabe „idiotensicher". Ein idiotensicheres Programm ist benutzerfreundlicher.

Internationales INPUT

J. Paul Igitty beschloß, ein eigenes Software-Haus (eine Firma, die Programme entwickelt) zu gründen, da die bestehenden nicht in der Lage waren, ein seinen Vorstellungen entsprechendes Programm zu schreiben. Einen Monat später nahm die „Igitty-Programme" ihre Arbeit auf.

Ihr erstes Produkt war eine **INPUT**-Routine, die eine J/N-Antwort vom Benutzer erfragt. Um aber auf dem internationalen Markt konkurrenzfähig zu sein, akzeptiert sie alle der folgenden Eingaben als "NEIN".

NEIN
NON
NO
NJET
NEE

Kannst Du das bewerkstelligen? Du kannst mehrere Bedingungen mit **OR** verknüpfen:

IF A$="NEIN" **OR** A$="NON" **OR** A$="NO" **OR** ...

Bedenke aber, daß auf dem Commodore 64 ein Befehl nicht länger als zwei Bildschirmzeilen sein darf. Vielleicht müßtest Du die Bedingungen in mehrere Befehle aufteilen.

Melanie Monsuns Melonen-Markt

Marta Monsuns Mutter Melanie baut Melonen an. Ihre Melonen, eine neue geschmacksintensive Züchtung, LANDMANNS STOLZ, zählen zu den besten in Lummoxshire. Sie stammen vom Minnesota-Melonen-Gürtel. (Die frühere Sorte, GELBE REVOLUTION, wurde in Nizza gezüchtet, wenn ich mich nicht irre.) Ein Problem gibt es jedoch mit Landmanns Stolz: Sie verträgt keine Temperaturen unter 13 °C und über 21 °C. Deshalb bezieht Melanie jeden Tag den Wetterbericht von ihrer Tochter Marta und läßt auf seiner Basis ein kleines Computerprogramm darüber entscheiden, ob sie das Fenster im Gewächshaus öffnen darf oder nicht. Die Fenster sollten nur geöffnet werden, wenn die Temperatur zwischen 13 °C und 21 °C (einschließlich) liegt.

Es gibt in BASIC leider keinen **ZWISCHEN**-Befehl (der dann wohl BETWEEN heißen müßte). Nach einem kurzen Gespräch mit Theodor Tastendrück, dem unvergleichlichen Programmierer, erfährt sie, daß ihre Temperatur TEMP, die zwischen 13° und 21° liegen muß, zwei Bedingungen erforderlich macht:

1. TEMP>=13
2. TEMP<=21

siehe Kasten auf Seite 73

Mit Hilfe des Befehlswortes AND (dem logischen UND) kann man feststellen, ob die Behauptung wahr ist:

IF TEMP>=13 **AND** TEMP<=21 **THEN** (was auch immer)

Bedingung 1 | Bedingung 2

Dank des folgenden Programms können jetzt Melanies maschinenbeschützte Melonen in den heftigsten Herbststürmen reifen:

```
10 PRINT"EINGABE DER TEMPERATUR"
20 INPUT TEMP
30 IF TEMP>=13 AND TEMP<=21 THEN
   PRINT"FENSTER OEFFNEN"
```

Proknatakadskis Code-Check

Genosse Sergei Proknatakadski, der russische Spion, wurde vom Kreml angewiesen, einen neuen Code zu verwenden. Der neue Code benötigt ein SCHLÜSSELWORT dieser Art:

RUMPELN

Es ist erforderlich, um die Buchstaben einer Nachricht zu verschlüsseln, indem das Alphabet wie folgt verschoben wird:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
R U M P E L N A B C D F G H I J K O Q S T V W X Y Z
```

Schlüsselwort | restliche Buchstaben in Reihenfolge

Das Schlüsselwort muß zwischen 6 und 11 Buchstaben aufweisen (da Moskau das als optimalen Bereich der Undechiffrierbarkeit des Codes bestimmt hat).

Der arme Proknatakadski hat aber nie gelernt, weiter als bis 5 zu zählen; er steckt also in ganz schönen Schwierigkeiten.

Kannst Du ein Programm schreiben, das ein eingegebenes Wort auf seine Verwendbarkeit als Schlüsselwort prüft und dann ausgibt "VERWENDE ES"? Bedenke, daß es zwischen 6 und 11 (einschließlich) Buchstaben lang sein muß. Verwende **LEN,** um die Länge des Wortes festzustellen. Das Programm sollte eine neue Eingabe zulassen, wenn die erste nicht den Anforderungen entspricht.

Antworten

Häßliche, grüne, verrückte Dinge.

Für häßliche, grüne, verrückte Dinge gibt es keine Antwort. Wir müssen lernen, mit ihnen zu leben.

Warum ist eine Maus, wenn sie sich dreht?

Tut mir leid, diesen Satz verstehe ich auch nicht.

Was ist der Unterschied zwischen einer Ente?

Ein Bein sieht aus wie das andere.

Aufgaben

(a) Frage, keine Behauptung.
(b) Wahre Behauptung.
(c) Unterhosen der Marke Honda; das ist aber keine Behauptung, das ist eine Frage.
(d) Wahre Behauptung.
(e) Datum, keine Behauptung.

(f) Wahre Behauptung. (Denn: Ich würde einen Regenschirm tragen, das kann ich Dir versichern.)
(g) Wahre Behauptung.
(h) Wahre Behauptung. (Heute ist Dienstag oder nicht.)
(i) Behauptung. Wahr nur, wenn Du wirklich der schwarze Mann bist **UND** ich meine fünf Mark fordere.
(j) Wahre Behauptung.
(k) Mittagessen, keine Behauptung.
(l) *Paradox.* Um wahr zu sein, müßte der Satz falsch sein. Wenn er aber falsch ist, muß er wahr sein. Am besten faßt Du ihn nicht als Behauptung auf (oder Dein Computer erleidet einen Nervenzusammenbruch).

Die Kartoffeln der Gräfin

„Es ist unmöglich, Holmes!"

„Unfug, Watson. In einigen Minuten erzählen Sie mir, wie offenkundig alles ist."

„So etwas würde ich nie über ein solch schwieriges Problem sagen, Holmes."

Der große Mann seufzte. „Beachten Sie, Watson, daß das Telegramm eine verknüpfte Behauptung ist, die aus fünf Teilen besteht:

1. Schweine haben Flügel, **UND** Königin Anne ist tot.
2. Der Frosch Kermit ist Präsident der USA.
3. Die preisgekrönte Kartoffel des Grafen wurde entführt.
4. Spinnen haben neun Beine.
5. „Chacun à son gout" ist französisch und bedeutet „Jeder hat noch einen gut".

„Davon ist Nummer (1) noch einmal selbst verknüpft. Beachten Sie weiterhin, daß die Behauptungen (1), (2), (4) und (5) offensichtlich FALSCH sind. Obwohl Königin Anne tot ist, sind Schweine nicht flugtauglich. Der Frosch Kermit ist nicht Präsident der USA."

„Da habe ich aber meine Zweifel, Holmes."

„Ich auch, aber wir schweifen ab. Spinnen haben acht Beine und nicht neun. Und ‚chacun à son gout' bedeutet ‚Jeder nach seinem Geschmack'."

„Brilliant, Holmes. Aber ...

„In der Tat. Aber: Wir wissen nicht, ob (3) wahr ist. Ich habe Sie jedoch gebeten, anzunehmen, die ganze Behauptung sei wahr. Eine verknüpfte Behauptung der Form:

(1) **ODER** (2) **ODER** (3) **ODER** (4) **ODER** (5)

kann aber nur wahr sein, wenn *mindestens ein Teil* wahr ist. Wir wissen schon, daß die Teile (1), (2), (4) und (5) falsch sind. Deshalb ..."

„Muß Behauptung (3) wahr sein!", rief ich aus. „Die Kartoffel des Grafen WURDE entführt! Sehen Sie es denn nicht, Holmes? Sie sind heute nicht in Form." Holmes setzte eine eigenartige Miene auf. Völliges Unverständnis. Ich versuchte, es ihm mit einsilbigen Wörter zu erklären: „Holmes, es ist so, wie Sie selbst sagten: ‚Wenn man das Unmögliche ausgeschaltet hat, dann muß das, was übrigbleibt - gleichgültig, wie unwahrscheinlich es ist - die Wahrheit sein!'"

Holmes lief rot an. Seltsam. Ein plötzlicher Hitzschlag? Erstaunen über meine messerscharfe Logik? Ich spürte das Verlangen, ihn wieder zu beruhigen.

„Warum, Holmes", fragte ich besorgt, „sehen Sie nicht, wie offenkundig alles ist?"

Internationales INPUT

```
10 PRINT"GEBEN SIE IHRE ENTSCHEIDUNG
   EIN: JA ODER NEIN"
20 INPUT E$
30 IF E$="NEIN" OR E$="NON" OR E$="NO"
   THEN GOTO 500
40 IF E$="NJET" OR E$="NEE" THEN GOTO
   500
...
500 (entsprechende Befehle ausführen)
```

Proknatakadskis Code-Check

```
10 PRINT"WORT EINGEBEN"
20 INPUT W$
30 LET L=LEN(W$)
40 IF L>=6 AND L<=11 THEN PRINT"VERWENDE
 ES"
50 IF L<6 OR L>11 THEN PRINT"GIB EIN NEU
ES WORT EIN"
60 IF L<6 OR L>11 THEN GOTO 10
```

9

Das INT und die Ameisenbären

Bernard Klöbner war in seinem Lehnstuhl eingenickt und wurde nun mit einem Mal aufgeschreckt. Seine Frau Ermintrude hatte ihr Studienbuch mit lautem Krachen auf den Boden geworfen. Sie sah sehr unglücklich aus; er wußte, daß sie mit ihren Nerven am Ende war.

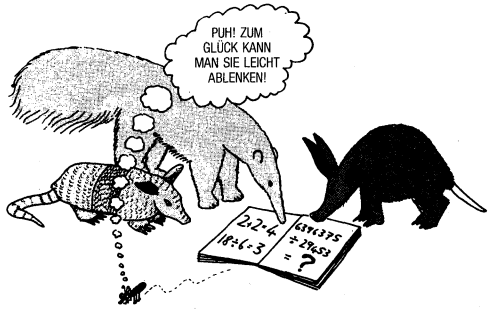
„Was ist los, Ermi?"

„Ach, Bernard", (sie sprach seinen Namen ‚Bernaah' aus), „es ist dieser gräßliche Korrespondenzkurs, den ich an der Offnungslos Universität belegt habe! Sieh Dir nur einmal diese Fragen an!"

Bernard hob das Buch, das grellgelb eingebunden war, vom Boden auf und sah auf den Titel:

AAAA 001: ARITHMETIK FÜR AMEISENBÄREN UND ANDERE AMEISENVERTILGER

Sehr sonderbar, dachte Bernard; aber schließlich war Ermi auch etwas sonderbar.



Vielleicht hatte sie es von ihrer Mutter geerbt, das Sonderbare. Er schlug das Buch auf einer von Ermi markierten Seite auf.

CKA 63: 29 435 Ameisenbären müssen sich 6 396 375 Ameisen teilen. Wieviele bekommt jeder? Wieviele bleiben übrig?

CKA 64: Ein Gürteltier geht in einen Laden und kauft 4 992 641 Doppelpaare Socken. (Beachten Sie: Gürteltiere haben vier Beine; einfache Sockenpaare reichen also nicht aus!) Wenn es jeden Tag ein Doppelpaar gebraucht und es dann wegwirft (was sehr vernünftig ist - haben Sie jemals die Socken eines Gürteltiers gerochen? Oder anders gefragt, haben Sie jemals ein Gürteltier selbst gerochen?), für wieviele Jahre reichen die Socken aus? Dabei können Sie die zusätzlichen Tage aus Schaltjahren vernachlässigen: Gürteltiere verschlafen den 29. Februar regelmäßig.

CKA 65: In einer Raumsonde umkreist ein Ameisenbeutler den Saturn einmal in 79 Tagen. Der Saturn umkreist seinerseits die Sonne in 10 757 Tagen. Wie oft umkreist der Ameisenbeutler den Saturn, während dieser einmal die Sonne umkreist?

Bernard dachte kurz nach. „Nun ja, ich würde sagen, das sind einfache Divisionsaufgaben."

„Ich weiß, Bernaah; aber Du weißt doch, wie schlecht ich im Dividieren bin! Ich habe ja schon Probleme, wenn ich bei Metzger Sandel die Rechnung nachrechnen will."

„Ich bin mir dessen durchaus bewußt, Ermi", seufzte Bernard. „Ähem. Was bedeutet CKA?"

„Computer-korrigierte Aufgabe, Liebling."

„Oh. Com - He! Ich hab's, Ermi! Warum benutzen wir nicht den Computer? Der Stork-37 kann schneller dividieren als du ‚Megaflop' sagen kannst!"

Und so kam es zu folgendem Programm

```
10 PRINT "CMA63:";6396375/29453
20 PRINT "CMA64:";4992641/365
30 PRINT "CMA65:";10757/79
```

Anzahl der Tage in einem Jahr

Sie ließen sich die Ergebnisse ausdrucken und schickten sie zur Korrektur.

Aber ...

... ganz so einfach war es auch wieder nicht. Zwei Wochen später kamen die Aufgaben zurück:

	FALSCH	RICHTIG
CMA 63: 217.172275	X	
CMA 64: 13678.4685	X	
CMA 65: 136.164557	X	

BEMERKUNG DES KORREKTORS: SIE HABEN DIE AUFFORDERUNG AUF SEITE 473 ÜBERLESEN, SIE ...

Nach kurzer Zeit hatten sie die Textstelle gefunden, auf die sich der Korrektor bezog:

Beim Beantworten der CKA 6Ø–8Ø ist jeder gebrochene Zahlenanteil zu ignorieren. Als Antworten gelten nur ganze Zahlen.

Das bereitete allerdings einige Schwierigkeiten, denn der Stork-37 schien nur in Dezimalbrüchen antworten zu können. Was sollten die Klöbners tun?

Integer

Schließlich fand Ermintrude Klöbner im Benutzerhandbuch des Stork-37 die Antwort. Das BASIC-Befehlswort

INT

wird dazu verwendet, eine Ausgangszahl in die größte ganze Zahl umzuwandeln, die kleiner oder gleich der Ausgangszahl ist. Das heißt, es läßt alles, was hinter dem Dezimalpunkt steht, weg (zumindest bei positiven Zahlen - bei negativen Zahlen arbeitet INT etwas anders). Zum Beispiel ist

INT(3.14159) = 3
INT(22.222222) = 22
INT(217.172275) = 217

Allgemein ergibt

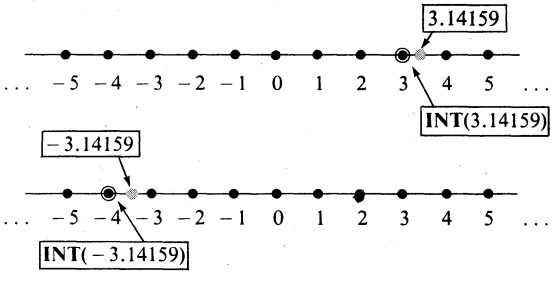
INT(Zahl)

den *ganzzahligen* Teil der Zahl (integer, engl.: ganze Zahl). Bei einer positiven Zahl ist dieser Teil das, was vor dem Dezimalpunkt steht. Bei einer negativen Zahl wird dieser Wert um Eins verringert (es sei denn, die Zahl ist bereits eine ganze Zahl). Versuche dieses Programm:

```
10 PRINT INT(3.14159)
20 PRINT INT(–3.14159)
30 PRINT INT(3)
40 PRINT INT(–3)
```

Du erhältst die Ergebnisse 3, –4, 3 und –3. Beachte, das **INT**(–3.14159) *nicht* –3, sondern –4 ergibt, weil die Zahl negativ und *nicht* ganzzahlig ist. **INT**(–3) dagegen ergibt –3, denn die Zahl ist zwar negativ, aber doch eine ganze Zahl.

Man versteht dies alles viel besser, wenn man den Zahlenstrahl betrachtet. Die Regel, daß **INT**(N) die größte ganze Zahl ergibt, die kleiner oder gleich N ist, läßt sich hier umformulieren zu: ergibt die größte ganze Zahl, die links von oder auf N liegt.



Programmiertechnisch gesehen ist INT kein Befehl, sondern eine FUNKTION: Wenn Du eine Zahl einsetzt, kommt auch eine Zahl heraus. Du kannst also Befehle wie

```
10 PRINT INT(10757/79)
20 LET Y = INT(12.35)
30 LET M(4) = 3*INT(Z/7)
```

geben, aber nicht

```
40 INT(10757/79)
```

Der Computer weiß nämlich nicht, was er mit dem **INT** machen soll, wenn er es errechnet hat.

> **INT** rundet eine Zahl auf die nächstkleinere ganze Zahl ab. Ist die Zahl bereits eine ganze, dann verändert **INT** nichts an ihr.

Die Klöbners versuchen es ein zweites Mal ...

Ermintrude und Bernard änderten ihr Programm ab:

```
10 PRINT "CKA 63:";INT(6396375/29453)
20 PRINT "CKA 64:";INT(4992641/365)
30 PRINT "CKA 65:";INT(10757/79)
```

Welche Ergebnisse erhalten sie?

... und schaffen es fast

Ihre Ergebnisse waren zwar richtig, aber sie hatten einen Teil der CKA 63 vergessen, die nicht nur nach

der Anzahl der Ameisen pro Bär, sondern auch danach fragte, wieviele Ameisen übrigblieben. Die Frage, die sie vergessen hatten, hieß also mit anderen Worten: Welchen Rest erhält man, wenn man 6 396 375 durch 29 453 dividiert?

Wir wollen das Problem etwas vereinfachen: Welchen Rest erhält man, wenn man 24 durch 5 teilt?

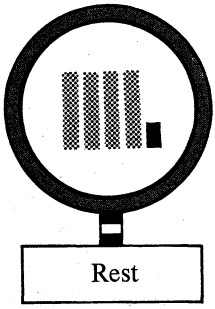
Nun, offensichtlich 4, aber wie können wir den Computer dazu bewegen, uns dies auszurechnen?

Eine Methode ist diese:

Schritt 1:	24 durch 5 dividieren (Ergebnis 4.8)	10 LET D = 24/5
Schritt 2:	den ganzzahligen Teil nehmen (Ergebnis 4)	20 LET I = INT(D)
Schritt 3:	diesen mit 5 multiplizieren (Ergebnis 20)	30 LET M = 5*I
Schritt 4:	das Ergebnis von 24 abziehen (Ergebnis 4, der gesuchte Rest)	40 LET R = 24–M

Verstanden? Wir errechnen, daß 5 viermal in 24 hineinpaßt. Das verbraucht 4*5=20, also sind noch 24–20=4 übrig.

Wenn wir das Programm betrachten, wird uns klar, daß wir es zu



10 **LET** R = 24 − 5 * **INT**(24/5)

kürzen können. Um es allgemein auszudrücken, ersetzen wir 24 durch N, 5 durch K und erhalten eine Formel für den Rest der Division N/K:

10 **LET** R = N − K * **INT**(N/K)

Hier ein einfaches Testprogramm für diese Formel:

```
10 PRINT"DIVIDENDEN EINGEBEN!"
20 INPUTN
30 PRINT"DIVISOR EINGEBEN!"
40 INPUTK
50 PRINT
60 PRINTN;"GETEILT DURCH";K;"ERGIBT";
70 PRINTINT(N/K);
80 PRINT"REST";
90 PRINTN-K*INT(N/K)
```

Du müßtest jetzt eigentlich den zweiten Teil von CKA 63 beantworten können.

Wieviele Ameisen bekommt man über, wenn man 6 396 375 in 29 453 Minuten hinunterwürgt? Oder habe ich da etwas durcheinandergeworfen?

> Wenn Du N durch K teilst, ist das Ergebnis
>
> **INT**(N/K)
>
> Der Rest beträgt
>
> N–K***INT**(N/K)

Neunzehnhundertvierundachtzig

... der Große Bruder sieht Dich an! Dabei gebraucht er natürlich hochkomplizierte computergestützte Überwachungsapparate. Aber der Große Bruder hat ein Problem. Er weiß, daß 1984 an einem Sonntag beginnt; sein Computer liefert ihm nette Informationen wie: Am 85. Tag des Jahres 1984 wurde Fred Pferdiger beim Lesen einer subversiven Illustrierten beobachtet. Sie trug den Titel „Das Freudenfest". Leider sagt der Computer dem Großen Bruder nicht, welcher Wochentag der 85. Tag des Jahres 1984 war. Dabei wäre dies besonders interessant: Es ist näm-

lich nicht nur legal, sondern sogar Pflicht, „Das Freudenfest“ an Dienstagen zu lesen, und selbst der Große Bruder will niemanden dafür bestrafen, daß er das Gesetz befolgt.

Kannst Du ihm ein Programm schreiben, daß die Tagesnummer (zwischen 1 und 366, 1984 ist ein Schaltjahr) als Eingabe annimmt, und ausgibt, welcher Wochentag der betreffende Tag ist?

Ein Tip: Betrachte den Rest nach einer Division durch 7.

Prim-Zeit

So, nun möchte ich, um dieses Kapitel mit einem Knalleffekt zu beenden, . . .

Kleinen Moment, es klopft jemand an der Haustür. Ich schaue nur kurz, wer es ist. Oh, nein! Ich glaub', es geht los! Fünf Mann in Stiefeln und Trenchcoats. Das hat man davon, wenn man Witze über den Großen Bruder reißt . . .

„Sie heißen Stewart?“

„Äh, . . . ja.“

„Wir sind von der BüsEC.“

„Bitte?“

„Der Organisation ‚Bürger für den sinnvollen Einsatz von Computern'."

„Oha."

„Wir sind darauf aufmerksam gemacht worden, daß dieses Buch eine unverhältnismäßig hohe Anzahl von Beispielen enthält, die alles andere als ernsthaft sind."

„Ja, natürlich, wissen Sie, ich versuche eben, unterhaltsame Beispiele zu bringen; aber wenn man hinter die Spaßigkeit schaut, dann entdeckt man immer auch einen durchaus ernsthaften Zweck, und jeder, der dieses Buch liest, wird viel grundlegendes Wissen . . ."

„Die BüsEC interessiert sich nicht dafür, was unter der Oberfläche liegt, Stewart! Wie alle Exekutiv-Gruppen haben wir schon genug Ärger mit dem Äußerlichen, ohne daß wir uns darüber Gedanken machen, was die Leute wirklich wollen."

„Es tut mir leid. Ich werde versuchen, meine Ziele offensichtlicher . . ."

„Nein, Sie werden mehr tun als nur bedauern! Sie werden eine wirklich sinnvolle Anwendung des Computers in Ihr Buch aufnehmen! Hier und jetzt!"

„Das lehne ich strikt ab! Das ist mein Buch, und ich schreibe, was ich will! Sie sind nur erfundene Figuren, ich kann Sie zwingen zu gehen, wann immer ich Lust dazu habe! Ich kann . . ."

Sie verschaffen sich mit Gewalt Zutritt in mein Haus. Es ist ein Alptraum. Ich scheine die Kontrolle über mein eigenes Buch verloren zu haben. Ich glaube, es gibt nur eine Lösung . . .

* * *

Oft kann man größere Zahlen erhalten, indem man kleinere Zahlen miteinander multipliziert. 72 ist zum Beispiel 3*24. Eine Zahl, die man auf solche Weise

erhalten kann, nennt man zusammengesetzt. Wenn eine Zahl nicht aus einem Produkt kleinerer Zahlen entstehen kann, wird sie als *Primzahl* bezeichnet. Die Zahlen

2 3 5 7 11 13 17 19 23 29

zum Beispiel sind die ersten zehn Primzahlen.

> Eine *Primzahl* kann nicht durch Multiplikation kleinerer ganzer Zahlen errechnet werden.

Das folgende Programm nimmt als Eingabe eine Zahl an und zerlegt sie in Primzahlfaktoren. Gib zum Beispiel 60 ein, dann lautet die Ausgabe:

60 = 2 * 2 * 3 * 5

Gibst Du 232841 ein, wird ausgegeben:

232841 = 7 * 29 * 31 * 37

Du wirst über die enormen arithmetischen Fähigkeiten Deines Computers erstaunt sein. (He, das geht leicht. Ich schieb' nur gerade mal einen Witz ein, während die Jungs nicht aufpassen. Kennst Du schon den von dem Erzbischof und der Banane? Also, er war glaube ich so . . . Aaaaaaaaaaahhh! Tut mir leid. Ich habe alles Menschenmögliche versucht.)

Hier das Programm.

```
10 PRINT"PRIMFAKTORIERUNG"
20 PRINT
30 PRINT"ZU FAKTORIERENDE ZAHL?"
40 INPUTN
50 PRINT
```

```
60 PRINTN;"=";
70 LETN0=N
80 IF2*INT(N/2)=NTHENPRINT" 2 *";
90 IF2*INT(N/2)=NTHENLETN=N/2
100 IF2*INT(N/2)=NTHENGOTO80
110 LETK=3
120 IFK*INT(N/K)=NTHENPRINTK;"*";
130 IFK*INT(N/K)=NTHENLETN=N/K
140 IFK*INT(N/K)=NTHENGOTO120
150 IFK*K>NTHENGOTO180
160 LETK=K+2
170 GOTO120
180 IFN=N0ANDN0>1THENPRINT" PRIMZAHL"
190 IFN<N0ANDN>1THENPRINTN
200 PRINT
210 PRINT
220 PRINT"DIESES PROGRAMM WURDE IM RAHME
N DER"
230 PRINT"ANTI-SCHMUTZ-UND-SCHUND-KAMPAG
NE DER"
240 PRINT"BUESEC MIT EINEM PREIS AUSGEZE
ICHNET."
```

Das Programm überprüft als erstes die Teilbarkeit durch 2. Dann nacheinander die Teilbarkeit durch die ungeraden Zahlen 3, 5, 7 . . . bis zur Quadratwurzel von N, über die man nicht hinausgehen kann.

Der nächste Abschnitt ist etwas schwieriger, und Du kannst wenn du willst, zu den Übungsaufgaben weiterblättern.

Wir wollen uns das Faktorierungsprogramm etwas näher anschauen. Um zu überprüfen, ob eine Zahl K der Teiler einer Zahl N ist, überprüfe, ob sich beim Teilen der Rest 0 ergibt. Wir wissen schon, daß der Rest durch die Formel

$$N - K*\mathbf{INT}(N/K)$$

gegeben ist. Der Rest ist also 0, wenn

$$K*\mathbf{INT}(N/K) = N$$

Die Programmzeilen 80 bis 100 prüfen also auf Teilbarkeit durch 2; 120 aus 140 prüfen auf Teilbarkeit durch K. Beachte, daß K in 110 den Anfangswert 3 erhält und in Zeile 160 jeweils um 2 erhöht wird zu 5, 7, 9, ... usw. Das Programm durchläuft eine Schleife, die durch die **GOTO**-Befehle bedingt ist und mit der abgesichert wird, daß alle Vielfachen von 2 überprüft wurden, bevor das Ganze mit 3 weitergeht, usw.

Die **IF**. . .**THEN**-Abschnitte der Zeilen 80 . . . 100 bzw. 120 . . . 140 sind gleich: Das kommt daher, daß wir DREI bestimmte Befehle ausführen lassen wollen, wenn die Bedingung erfüllt ist. Wir können nicht schreiben:

IF Bedingung **THEN** Befehl 1 **AND** Befehl 2 **AND** Befehl 3

Denn **AND** verknüpft nur Aussagen, keine Befehle! Wir können dies in drei Zeilen aufspalten:

IF Bedingung **THEN** Befehl 1
IF Bedingung **THEN** Befehl 2
IF Bedingung **THEN** Befehl 3

wobei wir jedesmal die gleiche Bedingung verwenden.

(Viele Computer, so auch der Commodore 64, erlauben auch *Mehrbefehlszeilen* der Form:

IF Bedingung **THEN** Befehl 1 : Befehl 2 : Befehl 3

Dies ist aber nur ein kleiner Seitenausgang, den ich hier nicht besprechen möchte.)

Um den Programmablauf noch besser zu verstehen, machen wir einen Trockenlauf. Als N wählen wir 2100:

Zeilennummer	N	K	N0	Bemerkungen
10-30	0	0	0	Benutzerhinweise
40	2100			Eingabe
50-60				Formatierung der Ausgabe
70			2100	Wert von N in einer Variable speichern, die nicht verändert wird
80				Aussage wahr: **PRINT** 2*
90	1050			Aussage wahr
100				Aussage wahr: **GOTO** 80
80				Aussage wahr: **PRINT** 2*
90	525			Aussage wahr
100				Aussage wahr: **GOTO** 80
80				Aussage falsch
90				Aussage falsch
100				Aussage falsch
110		3		
120				Aussage wahr: **PRINT** 3*
130	175			Aussage wahr
140				Aussage wahr: **GOTO** 120
120				Aussage falsch
130				Aussage falsch
140				Aussage falsch
150				Aussage falsch
160		5		
170				unbedingter Sprung: **GOTO** 120
120				Aussage wahr: **PRINT** 5*
130	35			Aussage wahr
140				Aussage wahr: **GOTO** 120
120				Aussage wahr: **PRINT** 5*
130	7			Aussage wahr
140				Aussage wahr: **GOTO** 120
120				Aussage falsch
130				Aussage falsch
140				Aussage falsch
150				Aussage wahr: **GOTO** 180
180				Aussage falsch
190				Aussage wahr: **PRINT** 7
200-240				Endmeldung

Wenn Du die äußerste rechte Spalte betrachtest, siehst Du, daß das Programm folgendes ausdruckt:

2100 = 2 * 2 * 3 * 5 * 5 * 7

Dieses Ergebnis ist richtig. Beachte, daß der Wert in der N-Spalte jedesmal abnimmt, wenn ein Faktor gefunden und ausdividiert wird.

Übungsaufgaben

Zerlege mit Hilfe des Programms die folgenden Zahlen in Primzahlfaktoren.

a) 60
b) 555555
c) 121771
d) 3778125
e) 11111117 (Der Commodore 64 benötigt zur Lösung dieses Problems mehr als 1,5 Minuten!)
f) 1024

In f) tritt ein sehr kleiner Programmfehler zutage. Kannst Du herausfinden, warum?

Ein weiteres ernstes Programm von der BüsEC

a) Ändere das obige Programm so ab, daß es in Form einer Schleife nacheinander die Primzahlenzerlegungen aller Zahlen N von 2 bis 10000 ausgibt. (Das dauert sehr lange; Du kannst daher auch nur die Zahlen von, sagen wir, 5000 bis 5500 ausdrucken.)
b) Schreibe ein Programm, das nacheinander alle Primzahlen ab einer gegebenen Zahl ausdruckt,

bis die größte vom Computer darstellbare Zahl überschritten ist.

Antworten

Die Klöbners versuchen es ein zweites Mal ...

CKA 63: 217
CKA 64: 13678
CKA 65: 136

CKA 63: Teil 2: 5074

Neunzehnhundertvierundachtzig

```
10 PRINT"DER GROSSE BRUDER SIEHT DICH AN
!"
20 PRINT"TAGESNUMMER EINGEBEN!"
30 INPUTT
40 IFT<1ORT>366THENGOTO20
50 LETR=T-7*INT(T/7)
60 IFR=1THENPRINT"SONNTAG"
70 IFR=2THENPRINT"MONTAG"
80 IFR=3THENPRINT"DIENSTAG, LEGALER FREU
DENFEST-TAG"
90 IFR=4THENPRINT"MITTWOCH"
100 IFR=5THENPRINT"DONNERSTAG"
110 IFR=6THENPRINT"FREITAG"
120 IFR=0THENPRINT"SAMSTAG"
```

Übungsaufgaben

a) 2 * 2 * 3 * 5
b) 3 * 5 * 7 * 11 * 13 * 37
c) 13 * 17 * 19 * 29

d) 3 * 5 * 5 * 5 * 5 * 5 * 13 * 31
e) Primzahl
f) 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2

Manchmal erscheint ein überflüssiges „*" am Ende – ein winziger Programmfehler. Harmlos, aber Du kannst ja trotzdem versuchen, ihn zu beseitigen.

Ein weiteres ernstes Programm von der BüsEC

a) Lösche Zeile 30 und 40 und füge folgendes hinzu:

```
30 FOR M = 2 TO 10000
40 LET N = M
195 NEXT M
```

b) Ein Lösungsweg ist, das alte Primfaktorierungsprogramm zu modifizieren. Ändere also Zeile 30 ab zu:

```
30 PRINT "ANFANGSZAHL FUER DIE
   PRIMZAHLENSUCHE?"
```

Lösche alle PRINT-Befehle, die Faktoren ausdrucken, jedoch nicht die Zeile, in der das Wort PRIMZAHL auftaucht. Du löschst also die Zeilen

60 80 120 190

und änderst Zeile 180 so ab, daß sie die Primzahl ausdruckt:

```
180 IF N = N0 AND N0 > 1 THEN PRINT N0
```

Endlosschleife:

```
185 LET N = N0 + 1
187 GOTO 70
```

Du wirst auch die GOTO-Befehle am Ende der Zeilen 100, 140 und 170 ändern müssen, denn die Zeilen 80 und 120 wurden gelöscht:

```
100 IF 2*INT(N/2) = N THEN GOTO 90
140 IF K*INT(N/K) = N THEN GOTO 130
170 GOTO 130
```

Diese etwas unsaubere Lösung ist außerdem sehr ineffektiv, denn der Computer führt immer noch die gesamte Faktorierung durch; er druckt sie nur nicht aus. In diesem Fall ist daher Faulheit eine Tugend. Die beste Lösung ist, das Programm neu zu schreiben, in ihm den Teiler 2 und die ungeraden Zahlen zu überprüfen und, wenn eine Primzahl gefunden ist, diese auszudrucken. Also etwa so:

```
10 PRINT"ANFANGSZAHL FUER DIE PRIMZAHLEN
SUCHE?"
20 INPUTN
30 IFN=1THENLETN=2
40 IFN=2THENPRINTN
50 IFN=2*INT(N/2)THENGOTO120
60 LETK=3
70 IFK*K>NTHENGOTO110
80 IFN=K*INT(N/K)THENGOTO120
90 LETK=K+2
100 GOTO70
110 PRINTN
120 LETN=N+1
130 GOTO50
```

Glossar

Absturz: Hat dann stattgefunden, wenn ein Programm an einer Stelle abbricht, an der es dies nicht soll. Zeichen für Fehler!

Akkumulator: Register der CPU, in dem unter anderem addiert wird.

Anführungszeichen: Ein Zeichen ("), das Ende und Anfang eines Strings anzeigt.

ASCII-Code: (American Standard Code for Information Interchange) repräsentiert Zeichen in standardisierter Form.

Ausgeben: Informationen für den Benutzer sichtbar machen.

Back-up: Kopie eines Programmes, die zur Sicherheit beiseite gelegt wird.

BASIC: (Beginner's All-purpose Symbolic Instruction Code) Bei Homecomputern sehr verbreitete Programmiersprache.

Bedingung: Eine Aussage, die entweder wahr oder falsch ist und von der die Verzweigung eines Programmes abhängig gemacht wird.

Befehl: Eine einzelne BASIC-Anweisung, wie zum Beispiel PRINT "HEINZ"

Befehlswort: Spezielles BASIC-Wort, wie PRINT, NEW, RUN, LIST.

Begrenzer: Unsinnige Eingabe, die als Signal für den Abbruch einer andernfalls endlosen Eingabeschleife dient.

Benutzer: Das bist Du.

Benutzerfreundlich: Als b. wird ein Programm bezeichnet, das vom Benutzer eine kurze Einarbeitungszeit fordert, hilfreiche Meldungen ausgibt, unterschiedlich formulierte aber inhaltlich gleiche Eingaben akzeptiert und dem Menschen statt dem Computer die Arbeit erleichtert.

Benutzer-Handbuch: Bedienungsanleitung für einen Computer.

Betriebssystem: Programm, das permanent im ROM des Computers gespeichert ist und der CPU sagt, wie sie sämtliche Routineaufgaben ausführen soll, die zum Betrieb des Computers notwendig sind.

Bildschirmspalte: Senkrechte Zeichenreihe auf dem Bildschirm.

Bildschirmspeicher: Bereich des RAM, in dem die auf dem Bildschirm erscheinenden Zeichen gespeichert werden.

Bildschirmzeile: Waagrechte Zeichenreihe auf dem Bildschirm.

Binär: Methode, Zahlen mit den Ziffern 0 und 1 darzustellen.

Bit: Die einzelne Stelle eines Codes; kann die Zustände 0 oder 1 annehmen.

Bug: (Wanze) Programmfehler.

Byte: Folge von acht Bits, zum Beispiel 10110010. Die meisten Homecomputer benutzen ein einzelnes Byte, um eine Informationseinheit zu speichern.

Chip: Ein elektronischer Schaltkreis, der mit photographischen Ätztechniken und Aufdampfungen auf einem winzigen Siliziumplättchen aufgebracht wird. Er vereinigt eine große Menge an Bauelementen auf kleinstem Raum.

Compiler: Ein sehr nützliches Programm, das dazu dient, ein BASIC-Programm in Maschinencode zu übersetzen und es so der CPU verständlich zu machen.

CPU: (Central Processing Unit) Das Gehirn des Computers, der Teil, der sämtliche Abläufe steuert. Besteht meistens aus einem einzigen Chip.

DATA-Liste: Folge von DATA-Befehlen in einem BASIC-Programm, die für das Programm benötigte Daten enthält.

Datei: Datenliste, die außerhalb des Computers gespeichert ist, zum Beispiel auf Kassette oder Diskette, und zur Verarbeitung innerhalb eines Programms wieder in den Computer eingelesen werden kann.

Daten: Informationen.

Debugging: (Fehlerberichtigung) Vorgang des Verbesserns eines fehlerhaften Programms.

Dezimalpunkt: Statt des Kommas steht in BASIC ein Punkt, also einhalb ist nicht „0,5“, sondern „0.5“.

Diskettenlaufwerk: Gerät, das Informationen in großen Mengen auf magnetisch beschichteten Kunststoffscheiben speichert.

Divisionszeichen: In BASIC steht nicht „:“, sondern „/“.

Dollarzeichen: ($) Wird in BASIC verwendet, um dem Computer zu signalisieren, daß er einen String behandeln muß, also eine Zeichenkette.

Drucker: Mechanisches Gerät, das eine gedruckte Ausgabe auf Papier erzeugt.

Eingabeschleife: Schleife, die einen Eingabebefehl enthält. Dient zur schnellen Eingabe großer Datenmengen.

Eingeben: 1. Über die Tastatur Informationen in den Computer eintippen. 2. Allgemein: Informationen in den Computer einfüttern.

Endlosschleife: Ein Zustand, in den der Computer manchmal gerät, meistens, wenn ein Programmfehler vorliegt. Er führt, ohne abzubrechen, immer wieder dieselben Befehle aus.

Faktorisierung: Darstellung einer Zahl als ein Produkt aus Primzahlen.

Fehlermeldung: Meldung, die der Computer ausgibt, wenn er auf einen Programmfehler trifft.

Feld: Variable mit listenähnlich numerierten Einträgen. Der N-te Eintrag in einem Feld mit dem Namen FELD wird unter FELD(N) angesprochen.

Flicken: Zusatz zu einem Programm, der durch Sprungbefehle angehängt wird, um einen Programmfehler zu beseitigen.

Flußdiagramm: Eine Diagrammform, die Kästchen und Pfeile verwendet, um den Programmablauf darzustellen. Als Lehrhilfe überholt, aber manchmal nützlich.

Funktion: In BASIC sind Funktionen Befehlswörter, die über festgelegte Rechenvorschriften aus gegebenen Werten neue errechnen. Sie sind keine eigenständigen Befehle.

Impuls: Signal im Innern des Computers, das zum Informationstransport dient.

Information: Tatsachen, Zahlenwerte, alles, was durch Symbole dargestellt werden kann.

Initialisieren: Am Anfang eines Programms oder eines Programmabschnitts Variablen Werte zuweisen.

Integer: Engl.: ganze Zahl. In BASIC ergibt die Funktion INT(N) die nächste Zahl, die kleiner oder gleich N ist. Beispiele: INT(6.83) ist 6, INT(6) ist 6, INT(–6.83) ist –7.

Interpreter: Programm, das der CPU BASIC-Befehle verständlich macht, indem es eine Zeile nach der anderen übersetzt.

Kassette: Speichermedium, auf dem Programme und Daten dauerhaft gespeichert werden können. Im Gegensatz zu Disketten billig, aber langsam.

Kilobyte: Einheit für die Speicherplatzkapazität. Entspricht 1024 Bytes, 8192 Bits oder auch grob gerechnet einer halben Textseite.

Laufvariable: Spezielle Variable, die sicherstellt, daß eine Schleife nach der gewünschten Anzahl von Durchläufen verlassen wird.

Logik: Die Lehre von der Ableitung beweisbarer Schlüsse aus beweisbaren Behauptungen. In der Computertechnik entspricht dies der Entscheidung zwischen WAHREN und FALSCHEN Aussagen.

Maschinencode: (Maschinensprache) Computersprache, die die CPU ohne Hilfsmittel versteht.

Mehrbefehlszeilen: BASIC-Programmzeile, die mehrere durch Doppelpunkt (:) getrennte Befehle enthält.

Mnemonic: (Hier:) Ein Variablenname, der an die Aufgabe der Variablen erinnert. Beispiele: PREIS, HOEHE, NAME$

Monitor: Bildschirm höherer Qualität, der zur Textausgabe verwendet wird.

Multiplikationszeichen: In BASIC ist dies weder „×“, noch „·“, sondern „*“.

Netzgerät: Gerät, das den Computer mit Strom der für ihn erforderlichen Spannung usw. versorgt. Bei Homecomputern ist es in der Regel vom eigentlichen Computer getrennt.

Null: Wird auf Computern als „Ø“ geschrieben, um sie vom Buchstaben „O“ zu unterscheiden.

Numerische Variable: Eine Variable, die Zahlen enthält.

Primzahl: Zahl, die nur durch 1 und sich selbst teilbar ist. Beispiel: 17.

Programm: Liste von Befehlen, die der Computer abarbeiten soll.

Programmspeicher: Bereich des RAM, der für Programme reserviert ist.

Programmzeile: In BASIC ist ein Programm in numerierte Zeilen gegliedert.

RAM: (Random Access Memory) Der Teil des Speichers, der von Programmierer verändert werden kann.

Register: Spezieller Speicherbereich innerhalb der CPU.

Rest: Bei einer Division die Differenz zwischen dem Dividenden und dem nächstkleineren Vielfachen des Divisors. Beispiel: 21/6=3 REST 3.

ROM: (Read Only Memory) Nur-Lese-Speicher. Wird von Computerherstellern verwendet, um dauerhaft Informationen, zum Beispiel das Betriebssystem, zu speichern. Kann nur durch Austauschen der Chips verändert werden.

Runtime error: Programmfehler, der sich erst zeigt, wenn die Programmzeile, in der er sich befindet, ausgeführt wird.

Schleife: Befehlsfolge, die mehrmals durchlaufen wird. Dabei werden in der Regel einige Variablenwerte verändert.

Schrittweite: Wert, um den die Laufvariable einer FOR ... NEXT-Schleife pro Durchlauf erhöht oder erniedrigt wird.

Software: Auf Kassetten, Disketten, EPROMs oder als Listings festgehaltene Programme.

Spalte: Senkrechte Zeichenreihe (auf dem Bildschirm).

Speicher: Elektronisches Bauteil, in dem der Computer Daten speichert.

Sprung: Befehl, der den Computer anweist, das Programm mit einer Zeile fortzusetzen, die nicht die in numerischer Ordnung folgende ist.

Spur: Im Englischen TRACE. Fachwort aus dem Bereich der Fehlersuche. Befehl, der anzeigt, welche Zeilen des Programms ausgeführt worden sind.

String: Bezeichnet jegliche Zeichenkette, auch die, die überhaupt kein Zeichen enthält.

String, leerer: Zeichenkette, die kein Zeichen enthält.

Stringvariable: Eine Variable, die einen String enthält.

Syntax error: Ein nach der BASIC-Grammatik falscher Befehl.

Tabulation: Übersichtliche Ausgabe in Spalten.

Tastatur: Wird vom Benutzer zur Dateneingabe verwendet.

Testzeile: Programmzeile, die zur Fehlersuche vorübergehend in das Programm eingefügt wird.

Textverarbeitungssystem: Intelligente Schreibmaschine, die Text verarbeiten kann, indem sie zum Beispiel auf Rechtschreibung prüft oder richtig trennt.

Trockenlauf: Methode der Programm-Fehlerbeseitigung, bei der das Programm zuerst von Hand abgearbeitet wird.

Uhr: Elektronischer Schaltkreis, der den zeitlichen Rhythmus aller Informationstransporte im Innern des Computers regelt.

Variable: Ein gekennzeichneter Speicherbereich, der eine Zahl oder einen String enthält und unter seiner Kennzeichnung gesucht werden kann. Die Kennzeichnung ist der Name der Variablen; er kann vom Programmierer innerhalb gewisser Grenzen frei bestimmt werden.

Variablenname: Codename, der einer Variablen gegeben wird, damit der Computer die einzelnen Variablen von einander unterscheiden kann. Als Beispiele: B, KRAFT, KATZE$

Variablenspeicher: Der Bereich des Speichers, in dem die Variablennamen und die zugehörigen Werte abgelegt werden.

Verkettung: Das „Aneinanderhängen" von Strings. Wird auch als Stringaddition bezeichnet. Beispiel "TISCH"+"LAMPE"="TISCHLAMPE"

Verzweigung: Befehl, bei dem der Computer zwischen verschiedenen Programmfortsetzungen entscheiden muß.

Wahrheitstafel: Tabellarische Darstellung aller möglichen Kombinationen von Wahr und Falsch zu einer logisch verknüpften Behauptung.

Wanze: (siehe *Bug*)

Wert: Die Information, String oder Zahl, die eine Variable momentan enthält. Kann vom Programmierer verändert werden.

Zeichen: Ein Symbol, das der Computer auf dem Bildschirm ausgeben kann, zum Beispiel X, 7, % usw.

Zeilennummer: Zahl, mit der jede Programmzeile beginnt. Wird vom Interpreter zur Orientierung im Programm benötigt.

Verzeichnis der Befehle

Befehlsworte

Mathematische Zeichen und der Umgang mit Strings

„Das Tor zur Computer-Welt – Commodore 64 für Fortgeschrittene" knüpft an die im Anfänger-Buch vermittelten Kenntnisse an:
Wer die grundlegenden Hürden beim Programmieren in BASIC erfolgreich genommen hat, lernt nun, wie man Daten speichert, Mehrfachschleifen zusammenstellt, Daten ändert und – wie Computer „denken".

Praktische Beispiele demonstrieren die Anwendung dieser Programmiertechniken: Wetterdatenanalyse, Primzahlberechnung, Telefonnummernverzeichnis.
Witz, Humor und vor allem Verständlichkeit zeichnen die Aufgabenstellung (jeweils mit Lösung) aus.

Eleanor Ball, Programmiererin, und *Ian Stewart*, Professor für Mathematik, setzen das programmiertechnische Wissen geschickt in unterhaltsame Geschichten um. Sie werden der spielerischen Absicht der Reihe „Das Tor zur Computer-Welt" gerecht: Programmieren lernen als Hobby mit Spaß!

ISB N 3-478-02250-9 DM +014.80

T 3-22-00 **mvg-Paperbacks**