



EINS

ODER

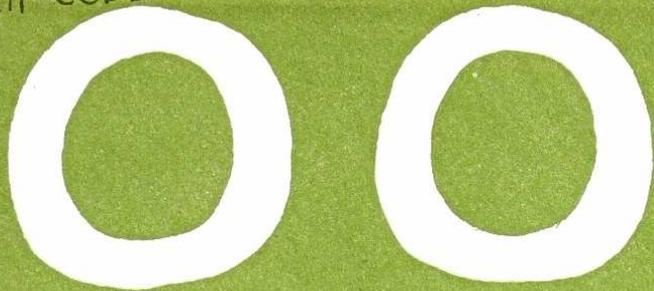
KEINS

COMPUTER

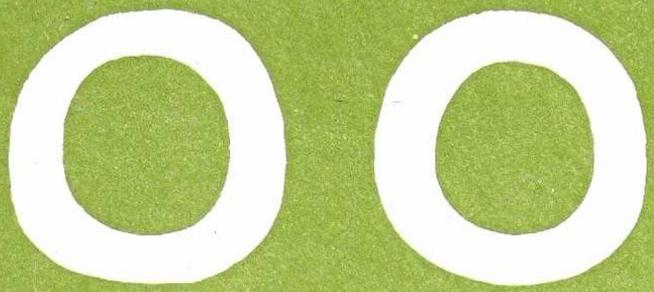
1 x 1

ASCII-CODE

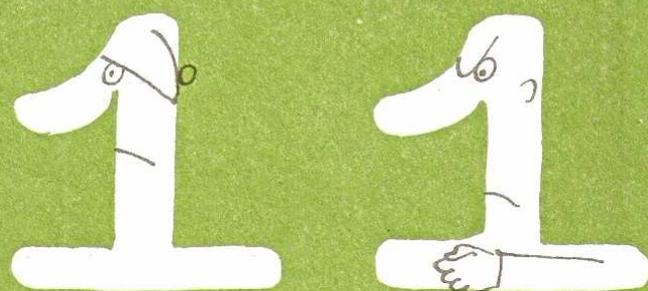
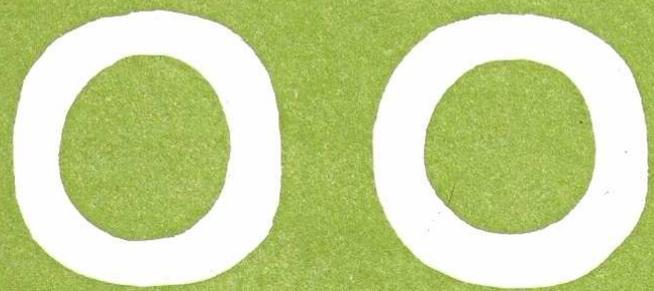
0



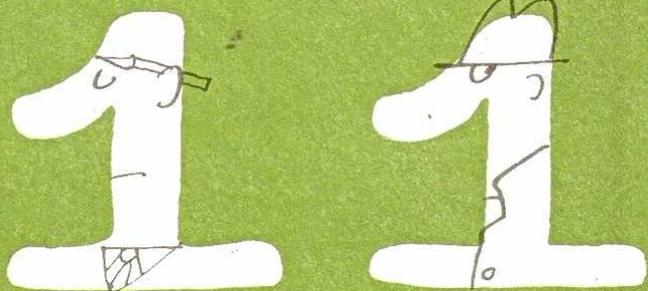
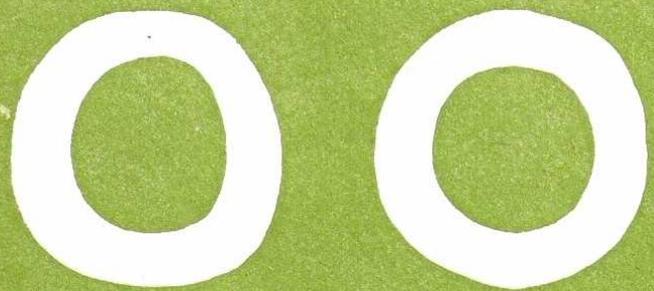
1



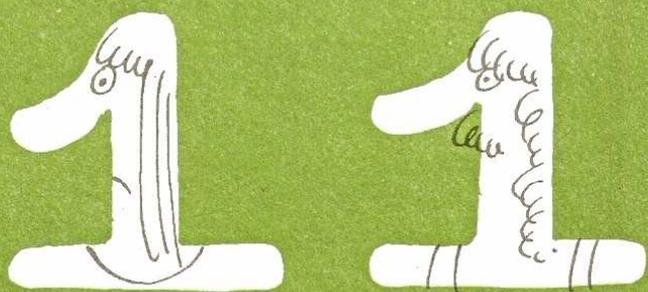
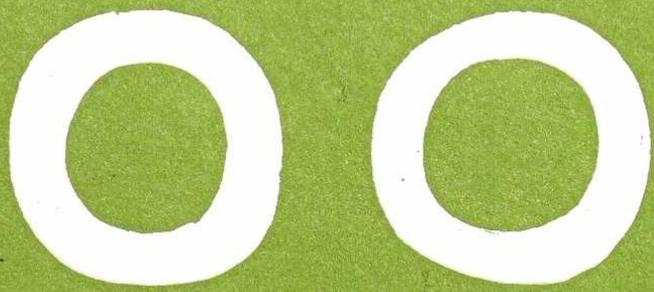
2



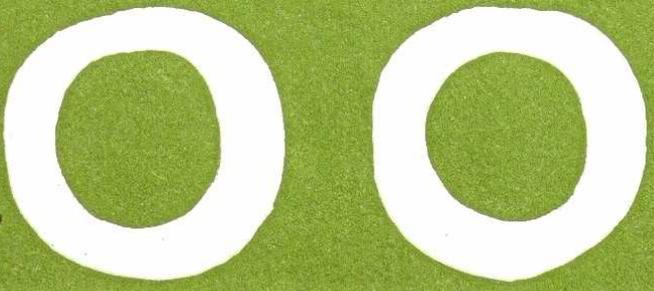
3

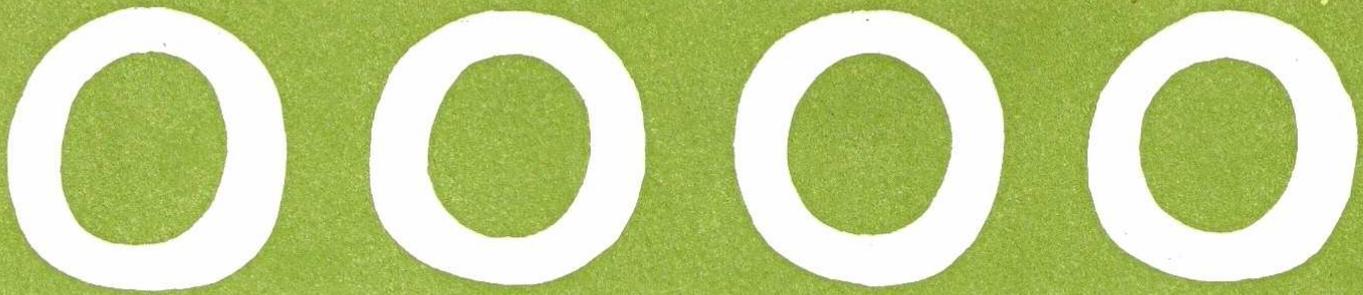


4

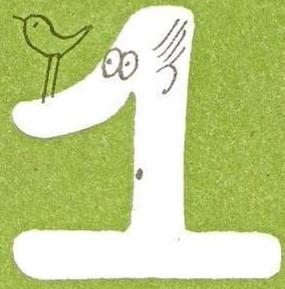
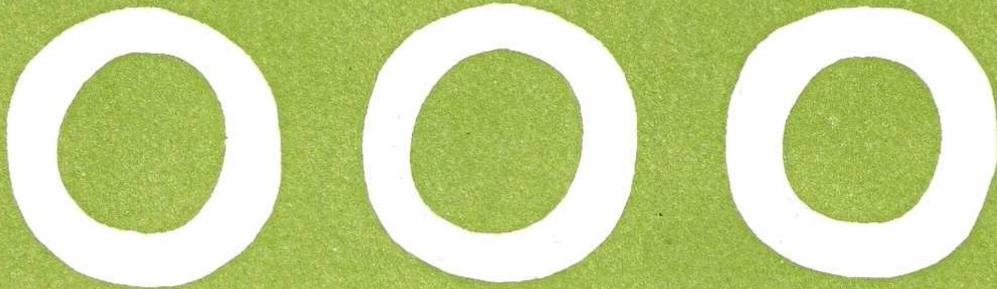


5

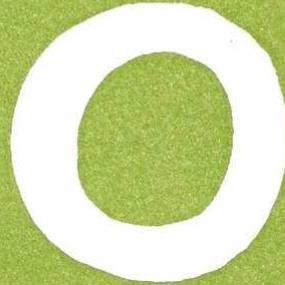




0



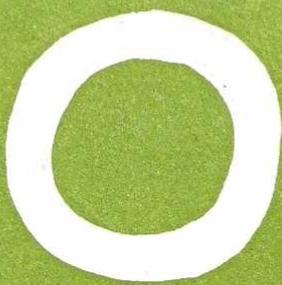
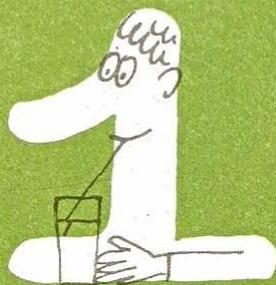
1



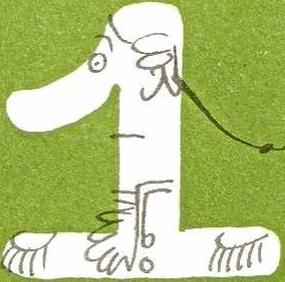
2



3



4

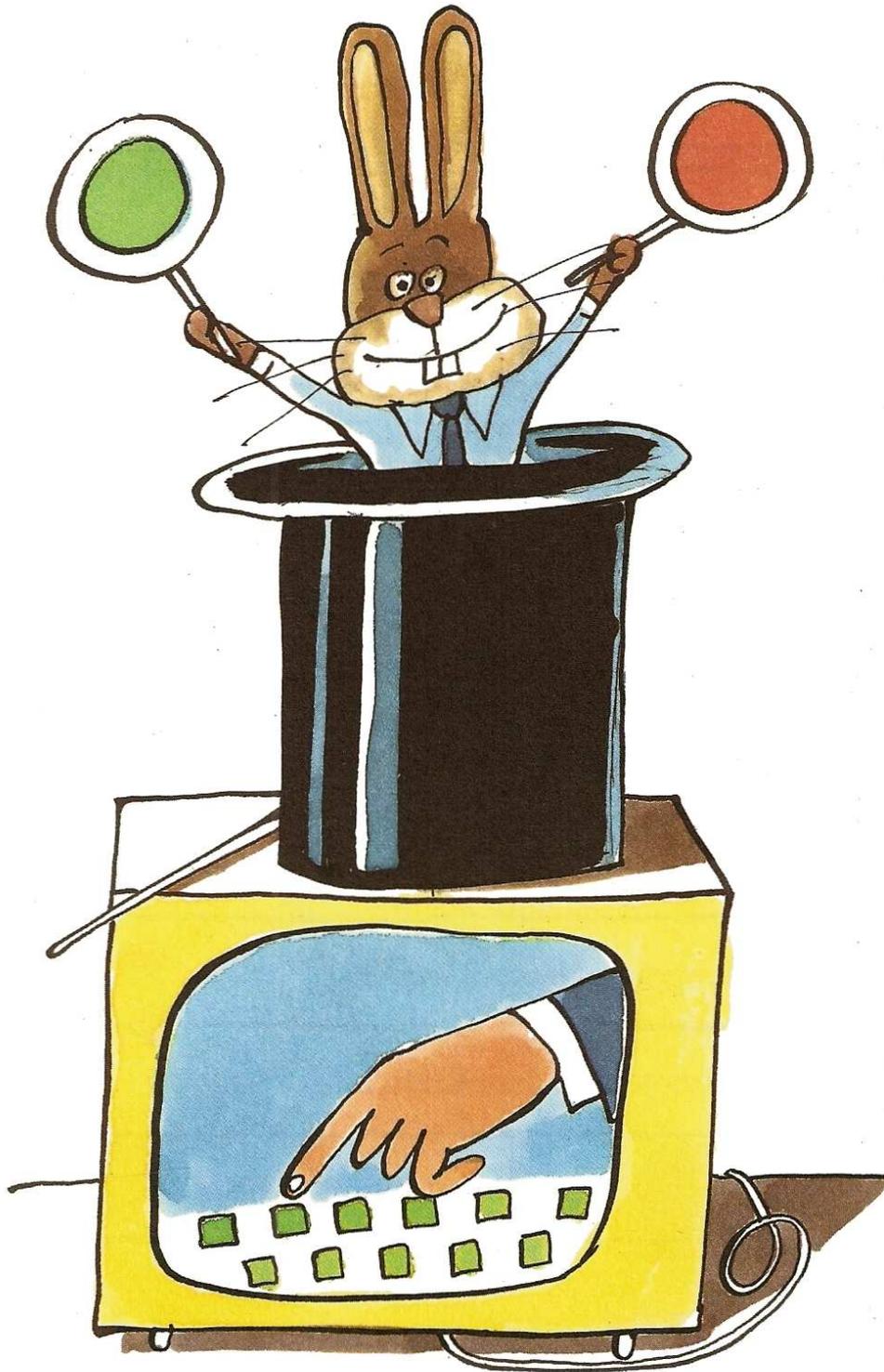


5

EINS ODER KEINS

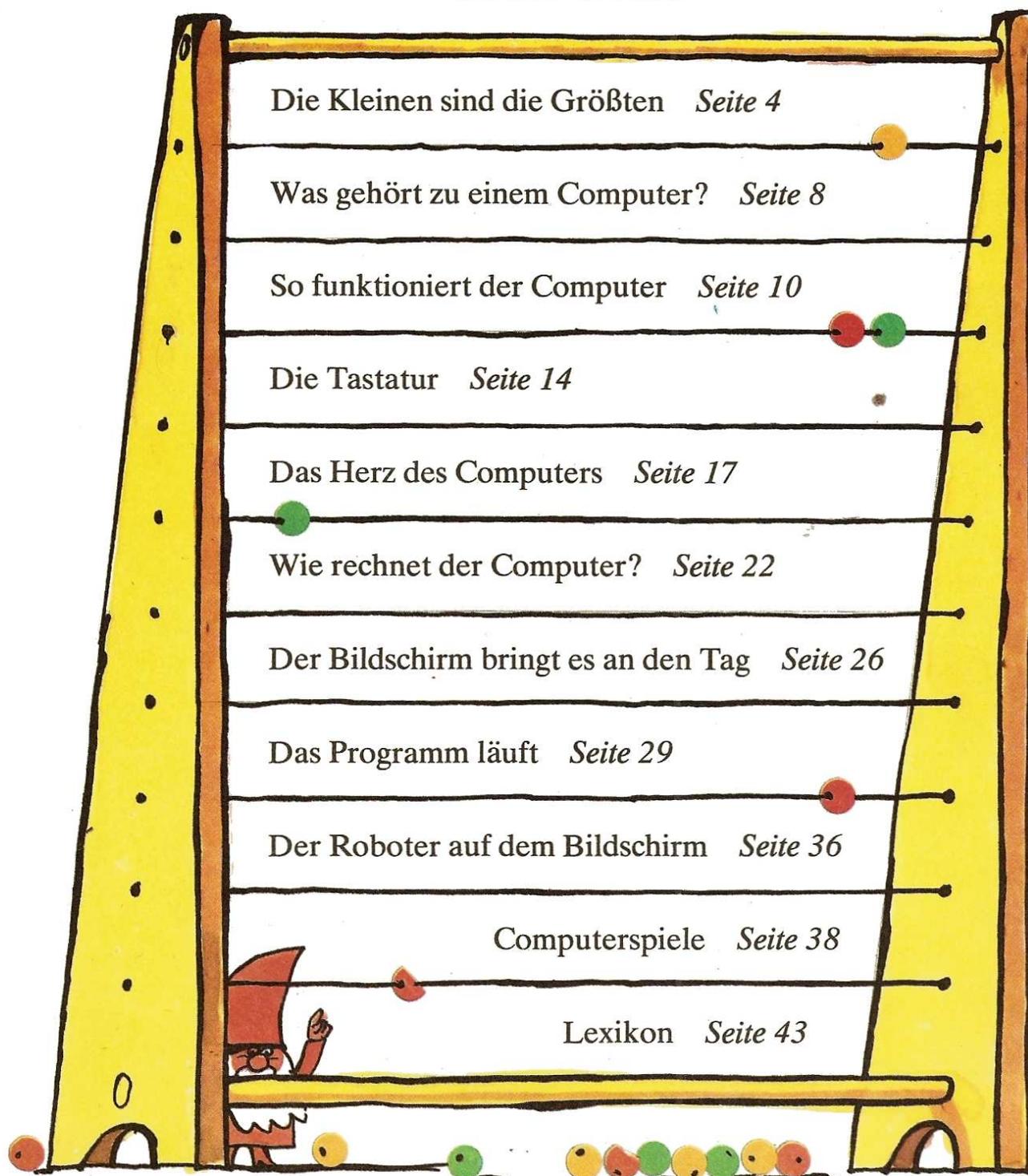
.....

Computer 1 x 1



Illustriert von Hans-Eberhardt Ernst

Inhaltsverzeichnis



Geschrieben von Hans Bäumler

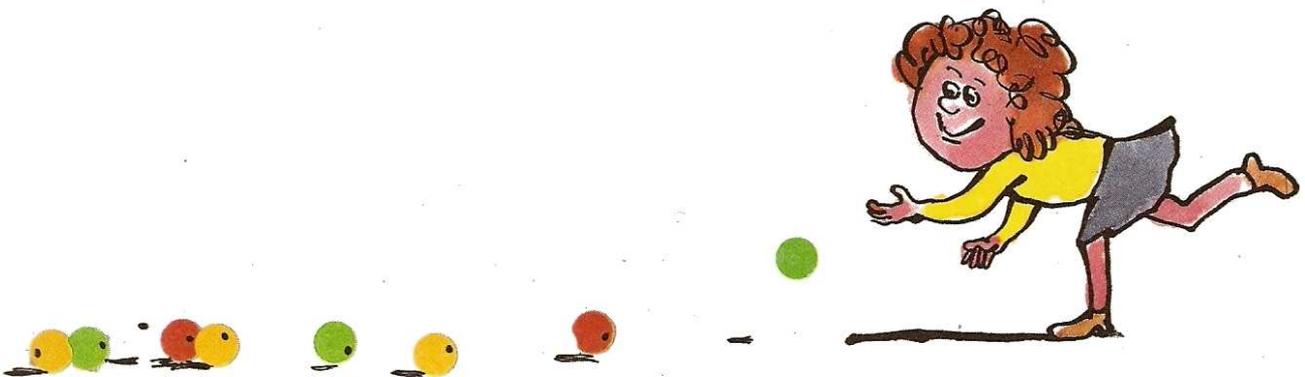
EINS
ODER
KEINS

.....

COMPUTER

1 x 1

Verlag Junge Welt



Die Kleinen
sind die Größten



Mikrorechnern begegnen wir heute schon fast täglich auf Bahnhöfen bei der Fahrkartenausgabe, in der Post oder in Sparkassen. Auch in Schulen und Pionierhäusern könnt ihr sie inzwischen finden. Kein Wunder, denn solch ein Computer ist ein geduldiger Lehrer, den nichts aus der Ruhe bringen kann. Im Fernsehen oder in Zeitungen habt ihr sicher schon eine Menge über Computer gesehen oder gelesen und bestimmt schon mal überlegt, ob nicht am besten ein Computer die verflixten Mathematikaufgaben gleich für die gesamte Klasse lösen sollte. Eigentlich wäre das kein Problem. Es fragt sich nur, wer den Computer unterrichtet. Wer erklärt ihm, wonach in den Textaufgaben gefragt ist? Und wer kontrolliert, ob die Ergebnisse richtig sind? Ihr habt es erraten. Ihr werdet das natürlich alles selbst tun. Aber der Computer kann euch dabei sehr gut helfen.

Beim Lesen dieses Buches werdet ihr feststellen, daß ein Computer nicht nur Mathematikaufgaben lösen kann. Wenn ihr gelernt habt, mit einem Computer umzugehen, wird er vielleicht zu eurem liebsten Spielzeug. Es dauert bestimmt nicht lange, und ihr werdet echte Computerfreaks (sprich frik; freak – begeisterter Mensch; eigentlich: Verrückter). Viele Wörter, die mit den Computern in Verbindung stehen, kommen aus dem Amerikanischen oder aus dem Englischen. Auf dem zum Buch gehörenden Lesezeichen findet ihr für solche Wörter Hinweise auf die richtige Aussprache.

Die kleinen Wunderkisten können natürlich noch viel mehr als nur rechnen oder spielen. Aus den Konstruktionsbüros der Betriebe oder den Forschungsinstituten der Universitäten und Akademien sind die Mikrocomputer schon lange nicht mehr wegzudenken. Komplizierte Berechnungen können in sehr kurzer Zeit durchgeführt werden.

Auf dem Bildschirm kann man Werkstücke konstruieren, Häuser oder Muster für Pull-over entwerfen. Das alles kann man sich dann auf Knopfdruck aus verschiedenen Blickwin-

keln ansehen. Außerdem lassen sich die Bilder noch vergrößern oder verkleinern. Der Konstrukteur wählt unter den vielen verschiedenen Entwürfen den besten aus und läßt den Computer gleich noch die Zeichnungen anfertigen. Dann kann im Betrieb mit der Produktion des neuen Gerätes begonnen werden. So hilft der Computer, viele Wochen Arbeitszeit einzusparen. Die Abkürzung CAD für diese Art des Konstruierens habt ihr bestimmt schon gehört (Computer Aided Design – computergestützte Konstruktion).

In den großen Betrieben wird der Computer nicht nur zur Konstruktion neuer Radios, Autos und Maschinen verwendet (CAD), sondern auch zur Herstellung dieser Produkte. Wird nach Anweisungen des Computers und unter seiner Steuerung produziert, dann benutzen wir die Abkürzung CAM (Computer Aided Manufacturing – computergestützte



Produktion). In manchen Betrieben werden die Computer durchgängig von der Konstruktion bis zur Produktion eines Erzeugnisses eingesetzt. Damit können in sehr kurzer Zeit viele neue Dinge preiswert hergestellt werden.

Seit einiger Zeit gibt es sogar Computer, mit denen man sprechen kann. Sie verstehen unsere Sprache und verarbeiten die von uns erteilten Anweisungen. Die ermittelten Ergebnisse stellen sie auf dem Bildschirm dar oder teilen sie uns über einen Lautsprecher mit. Diese Art der Verständigung mit dem Computer kann für kranke Menschen eine große Hilfe sein.

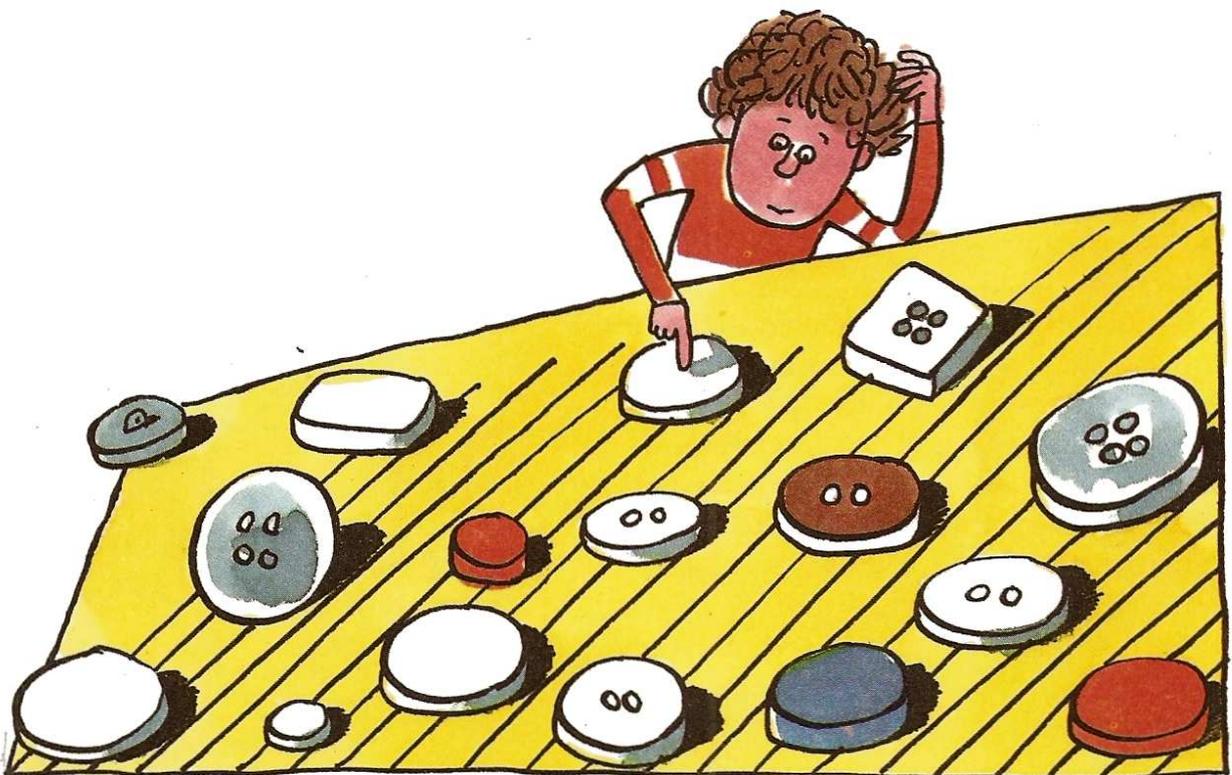
Auch beim Umgang mit sehr kleinen oder sehr großen Gegenständen, z. B. bei Arbeiten unterm Mikroskop oder an einem Hochofen im Stahlwerk, werden oft beide Hände benötigt. Dann ist ein Computer, der unsere Sprache versteht, sehr nützlich.

Solche Computer wären aber auch für die Übersetzung unserer gesprochenen Worte in eine andere Sprache geeignet. Vielleicht ha-

ben wir in nicht allzu ferner Zukunft bei Urlaubsreisen zu unseren Freunden ins Ausland einen kleinen Übersetzer im Ohr.

Unser Computer läßt sich auch als Schreibmaschine mit einem Riesengedächtnis verwenden. Viele eng beschriebene Textseiten lassen sich im Computer abspeichern. Der Computer sucht auf unseren Befehl hin jeden beliebigen Satz oder jedes beliebige Wort heraus. Er verschiebt ganze Kapitel, Abschnitte und Sätze an eine von uns gewählte Stelle. Er ersetzt aber auch einzelne Wörter oder Sätze durch andere. Das alles kann von uns auf dem Bildschirm kontrolliert werden, ohne daß ein Wort gedruckt oder neu geschrieben werden muß. Auf diese Weise lassen sich Bücher sehr effektiv schreiben. Auch das Manuskript zu diesem Buch wurde mit einem Computer geschrieben.

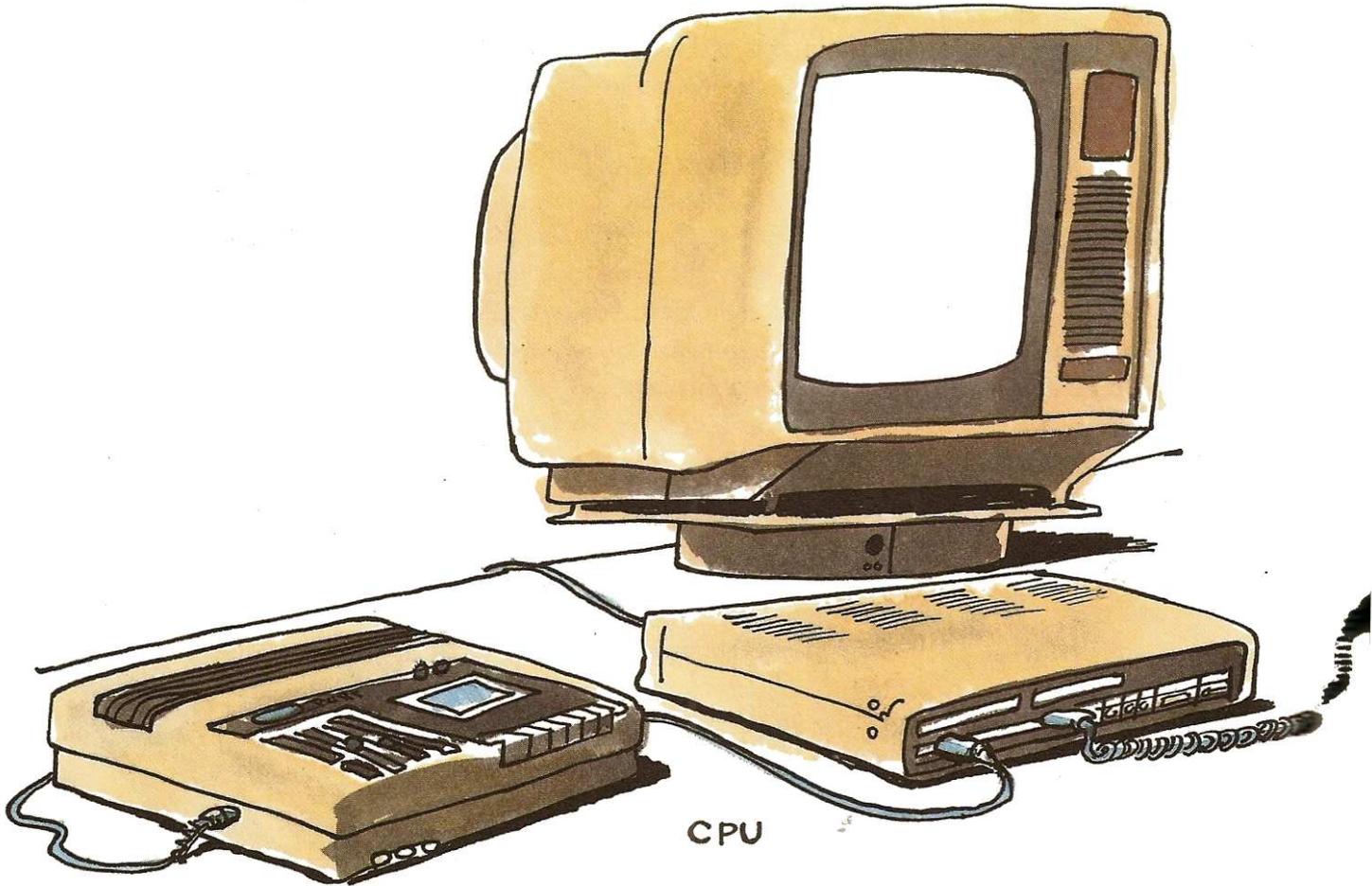
Wir haben hier nur einige Beispiele für den Einsatz der Mikrocomputer aufgeführt. Fragt doch einmal eure Eltern, wozu in ihren Betrieben die Computer verwendet werden.





Was gehört zu einem Computer?

BILDSCHIRM



Was den Computer zum Computer macht, ist nicht auf den ersten Blick zu erkennen. Zuerst fällt uns der Bildschirm auf, der wie ein Fernsehgerät aussieht. Dann die Tastatur, die an eine Schreibmaschine erinnert, aber viel mehr Tasten besitzt.

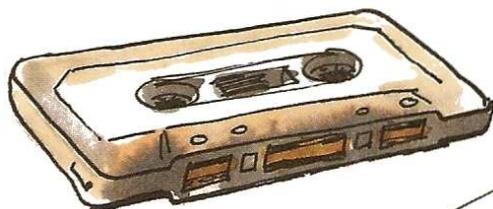
Die Tastatur brauchen wir, um dem Computer mitzuteilen, was er tun soll. Auf dem Bildschirm sehen wir dann, ob er uns verstanden hat. Alle Zahlen, Buchstaben oder Wörter, die wir eingetippt haben, werden auf ihm angezeigt. Aber nicht nur das. Auch die Ergebnisse, die aus dem Computer kommen, sind dort zu sehen. Das können Zahlen, Wörter, Striche und sogar Bilder sein.

Bei den meisten Computern können diese

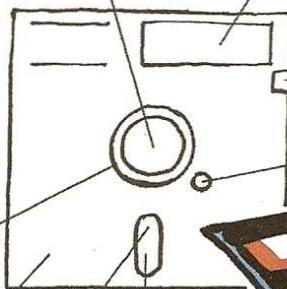
Ergebnisse außerdem noch auf einem Drucker ausgegeben werden. Auf diese Weise haben wir sie immer zur Hand, und der Computer muß sie nicht jedesmal, wenn wir sie benötigen, neu auf dem Bildschirm anzeigen.

Wo aber steckt nun der Computer? Meist ist er im Inneren des Gehäuses mit der Tastatur untergebracht und besteht aus vielen elektronischen Bausteinen. Der wichtigste Baustein ist die «zentrale Prozeßeinheit» CPU (Central Processing Unit – Zentrale Verarbeitungseinheit), der Mikroprozessor. Zu ihm gelangt, was wir über die Tastatur eingeben. Hier wird gerechnet und verglichen. Er steuert die Funktionen aller Bausteine und bestimmt auch, was auf dem Bildschirm erscheinen soll.

MAGNETBANDKASSETTE



Mittenloch Etikett



Schreibschutzkerbe

Indexloch

Antriebsring

Schutzhülle

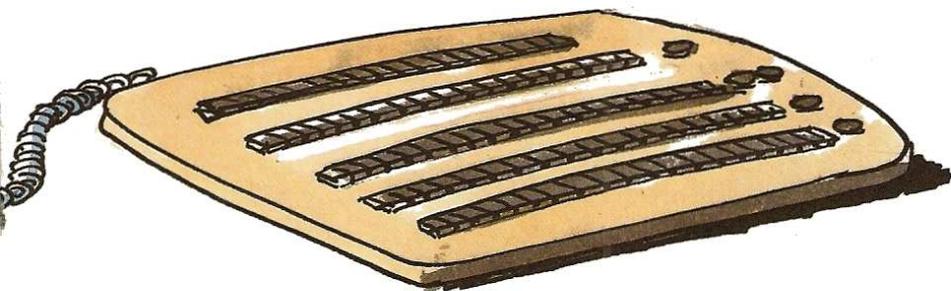
Kunststoffscheibe

Öffnung für
Schreib- und
Lesekopf



DISKETTE

TASTATUR



Damit die CPU richtig funktionieren kann, muß sie Anweisungen oder Befehle erhalten, z. B.: addiere, multipliziere, lies ein Wort, ziehe auf dem Bildschirm einen blauen Strich. Werden solche Befehle in einer bestimmten Reihenfolge angeordnet, sprechen wir von einem PROGRAMM. Wenn wir das Programm über die Tastatur eingetippt haben, wird es für den Mikroprozessor im ARBEITS-SPEICHER aufbewahrt. Es kann nun jederzeit von der CPU abgearbeitet werden. Dieser Speicher kann aber auch Zahlen oder Buchstaben (DATEN) aufnehmen und aufbewahren. Haben wir viele verschiedene Programme und Daten, dann ist es sehr mühsam, wenn man diese immer wieder über die Tastatur eintippen muß. Aus diesem Grund gibt es für den

Computer zusätzliche Speicher – Magnetbandkassetten oder kleine, wie biegsame Schallplatten aussehende DISKETTEN (Floppy disk – biegsame Scheibe), auf denen man sie, sind sie einmal geschrieben, abspeichern, also sichern kann.

An manche Computer kann man einen normalen Kassettenrecorder anschließen. Programme oder Daten werden dann vom Computer auf eine Kassette übertragen. Bei anderen Computern sind spezielle Diskettenlaufwerke oft gleich im Gehäuse der Tastatur oder unter dem Bildschirm untergebracht. Das Übertragen der Programme oder Daten auf die Diskette oder von der Diskette in den Computer geht hier viel schneller als beim Kassettenrecorder.

So funktioniert der Computer



Computer, Bildschirm, Kassettenrecorder oder Diskettenlaufwerk sind eingeschaltet. Wir sitzen vor der Tastatur, es kann losgehen. Drücken wir nun auf irgendwelche Tasten, passiert entweder gar nichts, oder es tauchen auf dem Bildschirm Zeichen auf, mit denen wir nichts anfangen können. Das ist nicht verwunderlich. Stellt euch vor, ihr drückt bei einer Schreibmaschine wahllos auf die Tasten. Da würde ein schöner Buchstabensalat entstehen. Aber auch wenn wir Worte schreiben wie: Der Ecke um sprang die Hund, ergeben sie erst einen Sinn, wenn sie in der richtigen Reihenfolge stehen, also: Der Hund sprang um die Ecke. Genauso verhält es sich beim Computer. Wenn wir wollen, daß er richtig

arbeitet, müssen wir ihm Schritt um Schritt mitteilen, was er tun soll.

Schauen wir uns ein kleines Beispiel an. Der Computer soll für uns die durchschnittliche Geschwindigkeit eines D-Zuges ausrechnen. Dazu müssen wir wissen, welche Zeit der Zug für eine bestimmte Strecke (Weg) braucht. Wir befehlen dem Computer:

```
10 LIES ZEIT
20 LIES WEG
30 RECHNE GESCHWINDIGKEIT =
   WEG/ZEIT
40 DRUCKE GESCHWINDIGKEIT
50 BEGINNE VON VORN
```

Das sind insgesamt 5 Befehle. Wir numerieren sie in Zehnerschritten, damit bei Bedarf neue Befehle eingefügt werden können, ohne daß man das Programm ändern muß. Nehmen wir an, daß der Zug für eine Strecke von 240 km eine Fahrzeit von 2 Stunden benötigt, dann rechnet uns der Computer eine Geschwindigkeit von 120 km pro Stunde (km/h) aus. Das Zeichen / in Zeile 30 benutzt der Computer für «geteilt durch». Wenn er das Ergebnis ausgedruckt hat, beginnt er von vorn und wartet auf neue Daten (Weg, Zeit). Die Ziffern am Beginn jeder Zeile bestimmen die Reihenfolge der Befehle. Eine so geordnete Folge von Befehlen ist ein PROGRAMM.

Es ist üblich, den Computern die Befehle nicht in deutscher Sprache zu erteilen. Wir benutzen eine speziell für den Umgang mit Computern entwickelte Sprache. Sie heißt BASIC. Da diese Programmiersprache recht leicht zu erlernen und weit verbreitet ist, werden wir uns nur mit ihr befassen. Es gibt aber auch noch andere Programmiersprachen, z. B. FORTRAN, COBOL, PL1 oder PASCAL.

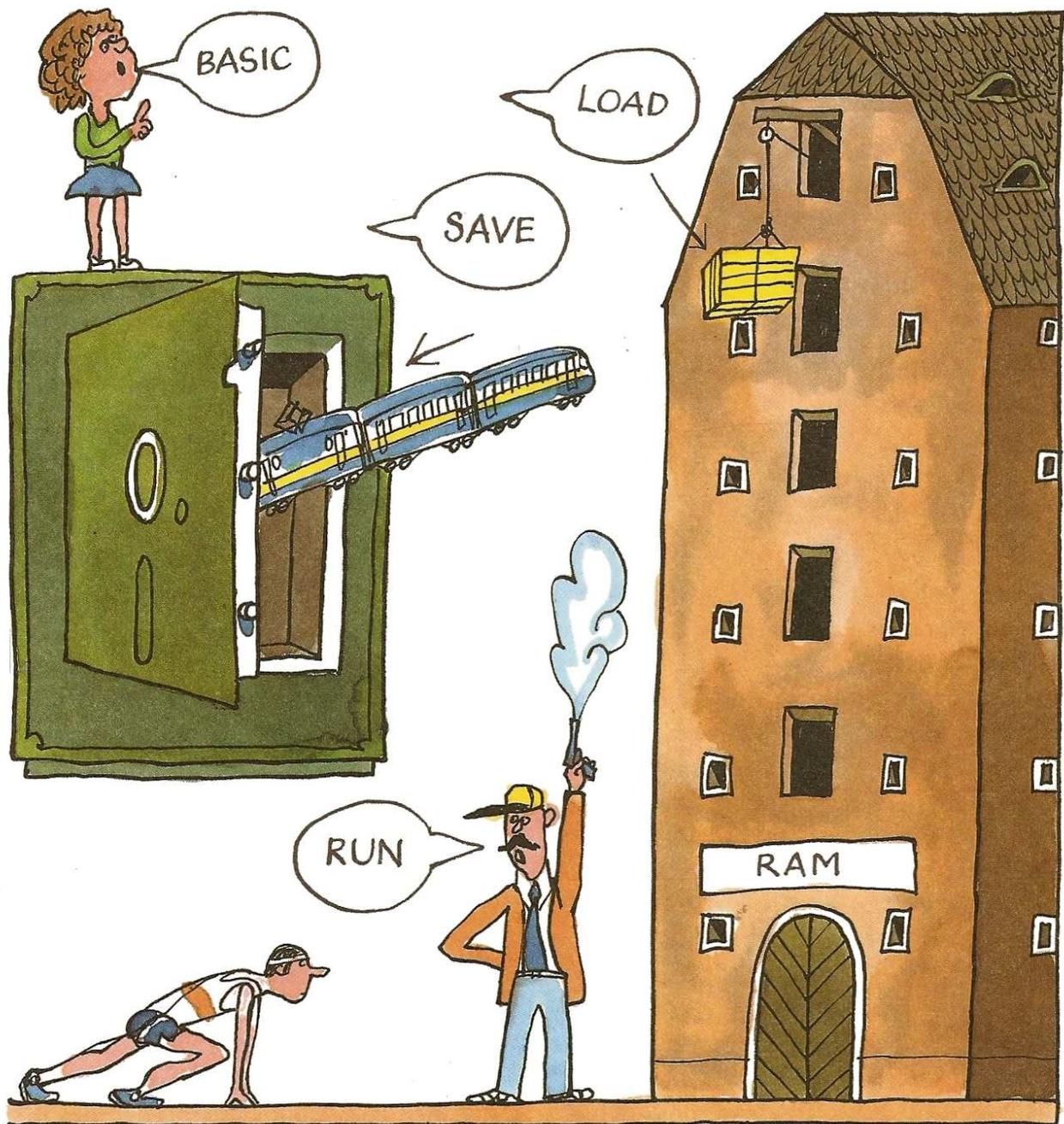
Die Basic-Befehle kommen aus dem Amerikanischen oder Englischen. Auf dem beiliegenden Lesezeichen findet ihr für jeden Befehl einen Hinweis, wie man ihn richtig ausspricht. Unser kleines Programm würde in Basic folgendermaßen lauten:

```
10 INPUT ZEIT
20 INPUT WEG
30 LET GESCHWINDIGKEIT =
   WEG/ZEIT
40 PRINT GESCHWINDIGKEIT
50 GOTO 10
```

Nachdem wir es eingetippt haben, können wir es auf der Diskette abspeichern. Damit wir es später rasch wiederfinden, geben wir ihm einen Namen. Wie wäre es mit ZUG? Mit Hilfe des Kommandos SAVE (save – sichern) wird das Programm aus dem Arbeitsspeicher des Computers auf die Diskette übertragen. Die Diskette können wir aus dem DISKETTENLAUFWERK herausnehmen und aufbewahren oder zu einem anderen Computer tragen. Unser Programm ist genauso wie ein Lied auf einer Magnetbandkassette sicher aufbewahrt. Das vollständige Kommando lautet: SAVE «ZUG».

Wollen wir am nächsten Tag unser Programm ZUG benutzen, brauchen wir es nicht erneut einzutippen. Mit Hilfe des Kommandos LOAD (load – laden) können wir es in den Arbeitsspeicher des Computers laden: LOAD «ZUG». Wie bei der Magnetbandkassette das Lied geht auch hier das Programm bei der Wiedergabe nicht verloren. Der Computer liest auf der Diskette wie in einem Buch und fertigt sich eine völlig identische Kopie des Programms an. Dieses befindet sich zwar nach dem Laden im Arbeitsspeicher, aber der Computer arbeitet es noch nicht ab. Er wartet darauf, daß wir ihm das Kommando RUN zum «Loslaufen» geben. Dann überträgt der Computer jeden Befehl unseres Programms erst einmal in seine eigene Maschinensprache und führt ihn aus. Die Übersetzung der Befehle aus der Basic-Sprache in die Maschinensprache führt der Mikroprozessor aus. Dazu benötigt er natürlich auch wieder ein Programm.

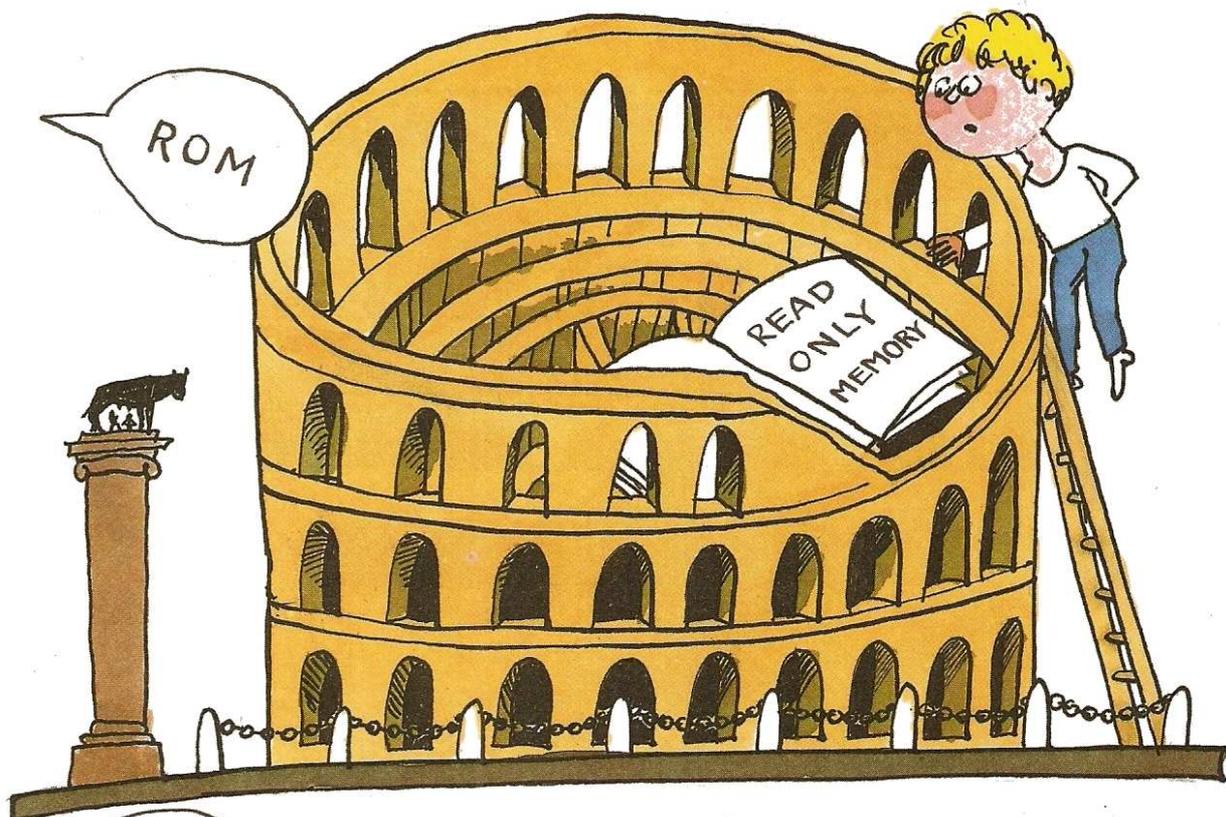
Bei vielen Computern wird es bereits vom Hersteller in einen Teil des Speichers geladen,



der besonders geschützt ist. Dorthin können keine anderen Daten oder Programme mehr abgelegt werden. Alles, was hier steht, kann nur gelesen werden. Deshalb bezeichnet man diesen Teil des Arbeitsspeichers als ROM (Read Only Memory; read only – nur zum Lesen, memory – Speicher). Der ROM ist mit einem Buch vergleichbar. Auch ein Buch kann immer wieder gelesen werden, ohne daß sich der Inhalt ändert, und ist nicht dafür gedacht, daß man etwas dazuschreibt. Ebenso ist es beim ROM. Denn neben dem Basic-Interpreter, so heißt das Übersetzungsprogramm,

gibt es im ROM noch viele kurze Programme für den Mikroprozessor. Mit diesen Programmen sorgt er dafür, daß alle Teile des Computers richtig zusammenarbeiten. Das eine hilft dem Mikroprozessor z. B., die eingetippten Befehle zu verstehen, ein anderes bewirkt, daß Buchstaben auf dem Bildschirm erscheinen. Alle diese kleinen, aber für den Betrieb des Computers sehr wichtigen Programme im ROM sind das BETRIEBSSYSTEM des Computers.

Damit dieses Betriebssystem weder absichtlich noch aus Versehen geändert werden kann,

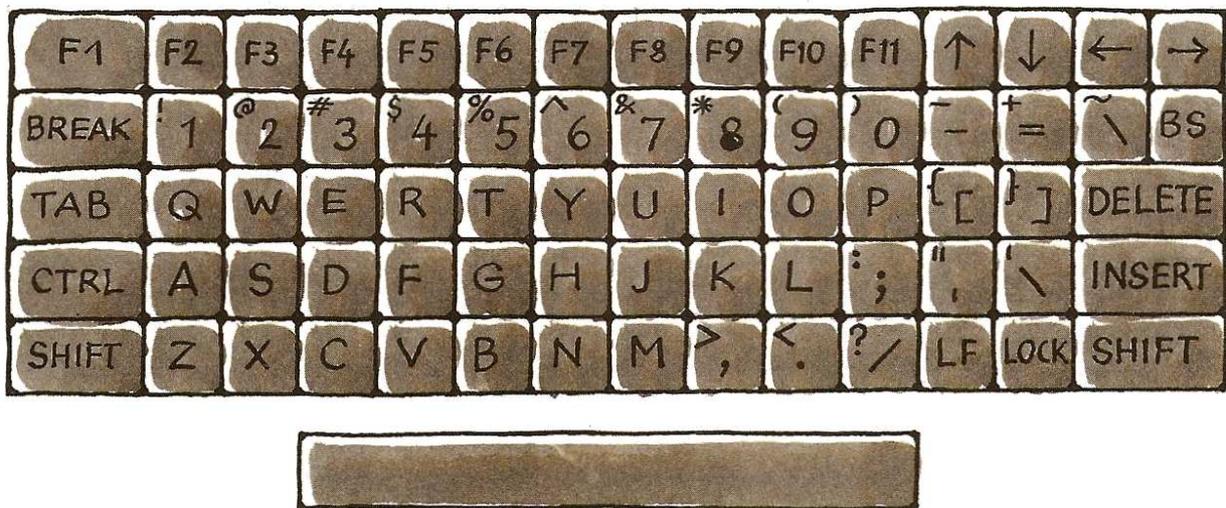


muß der Inhalt des ROM besonders gesichert sein. Er darf auch nicht gelöscht werden, wenn der Strom abgeschaltet wird. Schließlich sollt ihr euch gleich nach dem Einschalten durch Basic mit dem Computer verständigen können. Wäre der ROM nicht geschützt und wir drückten irrtümlich eine falsche Taste, könnte sein Inhalt leicht durcheinandergeraten. Und wäre es nur eine einzige Zahl, schon wäre das Betriebssystem verändert, und der Computer würde funktionieren, wie er will oder gar nicht mehr.

Es ist also sehr beruhigend, zu wissen, daß

wir dem Inhalt des ROM durch Irrtümer oder verrückte Programme keinen Schaden zufügen können. Dazu müßten wir unseren Computer mit Hammer oder Schraubendreher bearbeiten. Aber wer kommt schon auf so eine Idee?

Ganz anders verhält es sich mit dem restlichen, viel größeren Teil des Arbeitsspeichers. Hierhin wurde unser Programm ZUG geschrieben. Von hier aus wird es auch wieder gelesen. In diesen Speicher können wir sowohl schreiben als auch aus ihm lesen. Deshalb heißt dieser Teil des Arbeitsspeichers Schreib-



Lese-Speicher oder RAM (Random Access Memory: random access – beliebiger Zugriff, d. h. lesen oder schreiben). Schalten wir den Computer aus, werden alle Daten und Programme im RAM gelöscht. Sie sind für uns verloren. Deshalb Vorsicht! Wenn ihr ein neues Programm eintippt, solltet ihr unmittelbar nach dem Eingeben über die Tastatur euer Programm auf eine Diskette oder Kassette überspielen. Wenn jetzt der Computer, aus welchen Gründen auch immer, abgeschaltet wird, ärgert uns das nicht. Wir können das Programm ja ganz einfach wieder laden.

Wir wissen jetzt schon eine ganze Menge über den Computer:

- Das Programm schreiben wir für den Computer in einer besonderen Sprache. Sie heißt Basic.
- Das Programm gelangt von der Tastatur zum Schreib-Lese-Speicher. Das ist der RAM.
- Der Mikroprozessor erhält seine Anweisungen aus Programmen des Betriebssystems im ROM.
- Das Programm zum Übersetzen der Basic-Befehle in die Sprache des Computers befindet sich meist schon im ROM. Es ist der Basic-Interpreter.
- Jedes Programm wird vom Mikroprozessor Schritt für Schritt abgearbeitet.

Die Tastatur

Wir haben die Tastatur zwar schon benutzt, wollen sie uns aber noch etwas näher betrachten. Die Eingabe der Befehle, Daten und Kommandos erfolgt Zeichen für Zeichen, wie bei einer Schreibmaschine. Auf der Tastatur findet ihr das gesamte Alphabet, die Ziffern 0 bis 9 und sogenannte Sonderzeichen, wie z. B. ! ? , . ; : &. Einige dieser Sonderzeichen sind für uns besonders wichtig, da wir sie zum Rechnen brauchen: plus +, minus –, mal \times , geteilt durch / und die Klammern (). Aber auch größer als >, kleiner als <, größer oder gleich \geq , kleiner oder gleich \leq und ist gleich = versteht der Computer. Außerdem gibt es noch Funktionstasten, z. B. zum Potenzieren oder für spezielle mathematische Funktionen, Tasten zum Auswählen von Farben auf dem Bildschirm, eine Taste zum Umschalten von Groß- auf Kleinschreibung (SHIFT) und eine Taste BREAK zum Unterbrechen eines laufenden Programms (break – unterbrechen). Die Taste ENTER (enter – hineingehen) benutzen wir jedesmal, wenn wir mit der Eingabe einer Befehlszeile oder von Daten fertig sind, also auch nach RUN, SAVE, LOAD usw. oder nach INPUT-Anweisungen in Programmen, wenn wir unsere Eingabe gemacht

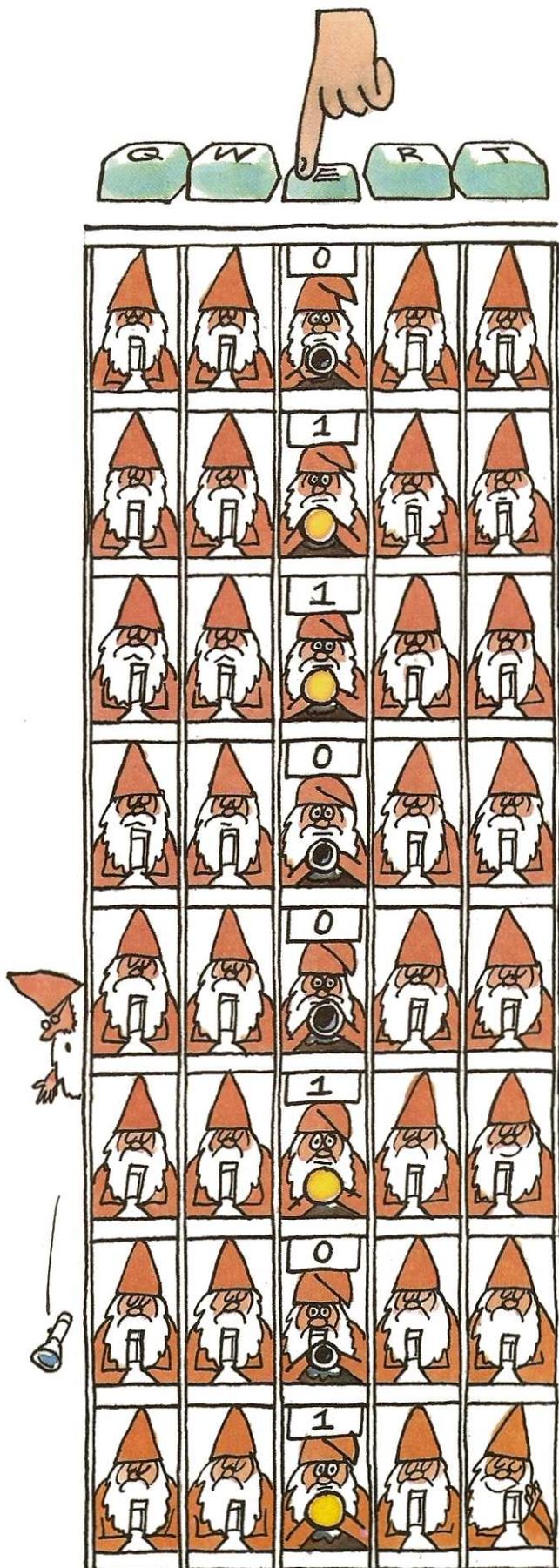


REISE

RI:SE

RIESE





haben. Dadurch gelangt alles, was ihr eingetippt habt, in den RAM. Vertippt ihr euch einmal – und das geht schneller, als man denkt –, dann helfen euch die Tasten für den CURSOR (Positionsanzeiger). Das sind vier mit Pfeilen \uparrow \downarrow \rightarrow \leftarrow gekennzeichnete Tasten. Durch Drücken dieser Tasten gelangt ihr mit dem Cursor an jede beliebige Stelle eures Textes auf dem Bildschirm. Die Position, an der sich der Cursor befindet, erkennt ihr an einem blinkenden Viereck. Habt ihr die Stelle, wo der falsch eingegebene Buchstabe steht, erreicht, drückt ihr die Taste DELETE (delete – löschen). Schon ist der Fehler weg. Jetzt braucht ihr nur noch den richtigen Buchstaben einzutippen. Bei manchen Computern muß man vor dem erneuten Eintippen die Taste INSERT (insert – einsetzen) drücken.

Die Tasten für den Cursor werdet ihr sicher noch sehr oft benutzen, denn bei vielen Computerspielen werden damit die Figuren bewegt, wenn sie durch Irrgärten eilen oder Abenteuer zu bestehen haben.

Was passiert nun eigentlich, wenn wir auf eine Taste drücken? Bei einer Schreibmaschine sieht man, wie die Typen durch einen kleinen Hebel oder durch einen kleinen Hammer auf das Papier gepreßt werden. Bei der Tastatur des Computers gibt es so etwas nicht.

Der Druck auf die Tasten erzeugt elektrische Impulse, die zum Mikroprozessor geleitet werden. Dabei spielt es überhaupt keine Rolle, wie stark dieser Druck ist. Die Impulse ändern sich dadurch nicht. Daran solltet ihr ab und zu denken, wenn ihr beim Spiel ein Auto auf dem Bildschirm durch die Kurven flitzen laßt. Es genügt ein leichter Druck auf die Tasten, und schon gehen jedesmal acht Impulse auf die Reise. Das sind bei jeder Taste natürlich andere, damit der Mikroprozessor erkennen kann, welche Taste gedrückt wurde. Man könnte fast denken, daß in jeder Taste ein Männlein sitzt, das zu genau festgelegten Zeiten einen Schalter drückt oder losläßt – der Schalter ist also ein- oder ausgeschaltet.

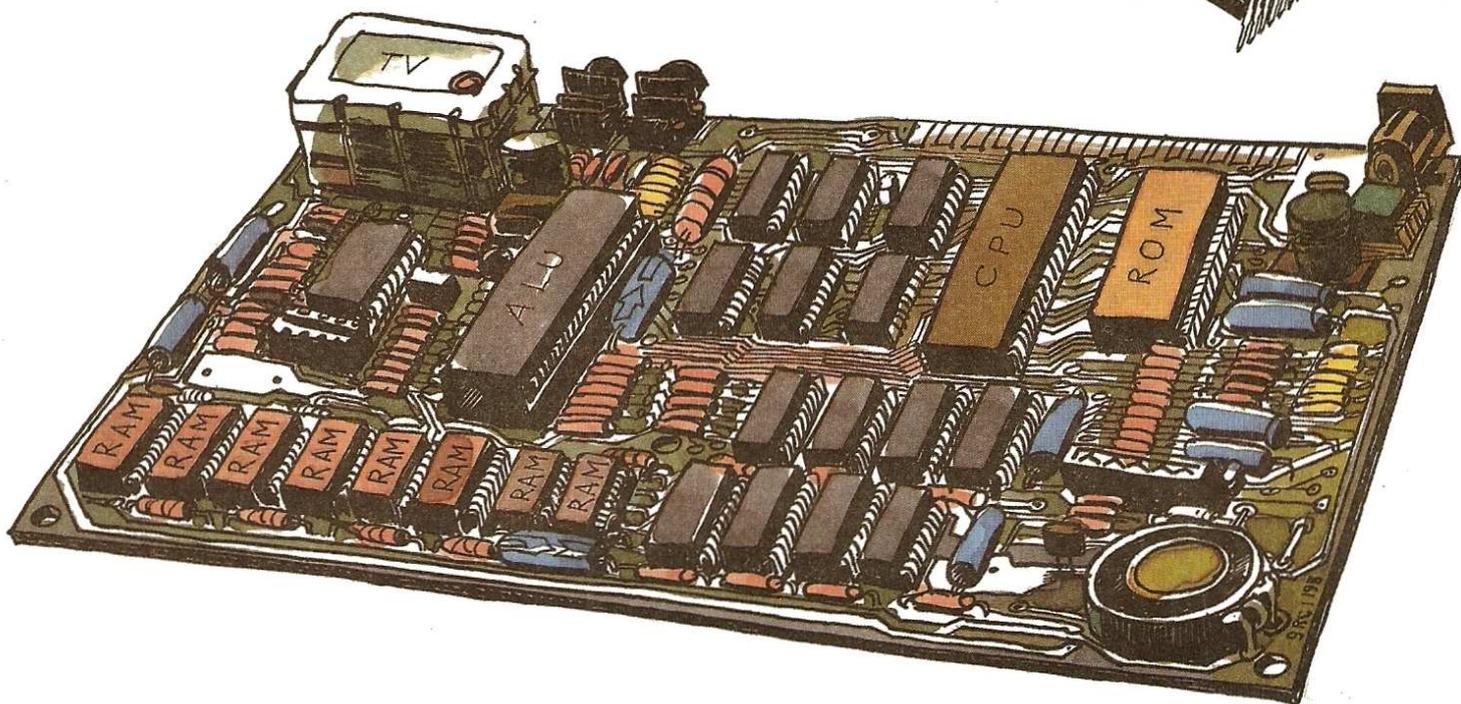
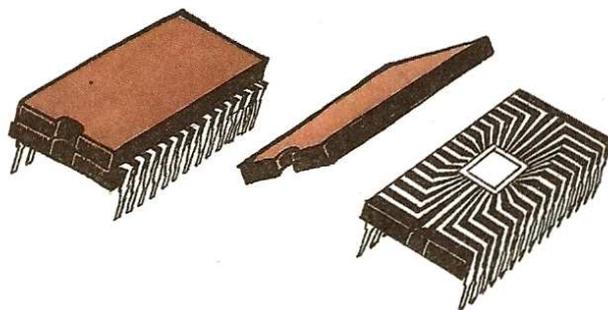
Drückt ihr die Taste mit dem Buchstaben E, würde es die Impulsfolge «aus – ein – ein – aus – aus – ein – aus – ein» erzeugen, und bei der Ziffer 3 wäre es die Folge «aus – aus – ein – ein – aus – aus – ein – ein». Schreiben wir für «aus» eine 0 und für «ein» eine 1, sind diese Folgen viel übersichtlicher: E = 0 1 1 0 0 1 0 1 und 3 = 0 0 1 1 0 0 1 1. Den Schlüssel, um aus Buchstaben, Zahlen und Sonderzeichen die richtigen Impulsfolgen machen zu können, liefert der ASCII-Code. Dieser Code wird bei allen Computern verwendet, die mit BASIC arbeiten. Aber auch andere, viel größere Computer benutzen ihn zur Dateneingabe.

Mit den acht Impulsen können 256 Zeichen verschlüsselt werden, denn so viele verschiedene Kombinationen von 1 und 0 gibt es. Wollt ihr es nachprüfen? Dann versucht euch erst an vier Impulsen. Da erhaltet ihr nämlich nur 16 verschiedene Möglichkeiten. Für den Fall, daß die Folgen nur aus drei Impulsen bestehen würden, seht ihr die 8 verschiedenen Möglichkeiten in der Tabelle, wobei wir bei Null mit dem Zählen beginnen:

Folge	Codierung	Folge	Codierung
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

Das Herz des Computers

Nachdem wir uns die Tastatur gründlich von außen angeschaut haben, wollen wir sehen, was es im Inneren gibt. Öffnen wir das Gehäuse der Tastatur, liegt eine dünne Platte von der Größe eines Schulheftes vor uns. Auf dieser Platine oder Leiterplatte befindet sich die gesamte Elektronik des Computers. Die elektronischen Bausteine erinnern an Maikäfer. Sie haben nur viel mehr Beine, die hier PINS (pin – Stift) heißen. Ein besonders stattlicher Maikäfer mit 40 Pins ist der uns schon bekannte Mikroprozessor. Auch die großen RAM und der kleinere ROM des Arbeitsspeichers sind zu erkennen. Die schmalen, länglichen Kunststoffklötzchen mit den kleinen Löchern sind Steckleisten. Sie können weitere Bausteine aufnehmen, ohne daß auf der Platine oder an den Bausteinen gelötet werden muß. Will man den Arbeitsspeicher erweitern, wird einfach ein RAM-Baustein aufgesetzt. Es gibt aber auch andere Erweiterungsmodule,



wie ROM-Bausteine oder eine weitere CPU.

Sowohl auf der Ober- als auch auf der Unterseite der Platine sehen wir Verbindungslinien zwischen den Bausteinen. Sie sind elektrisch leitfähig. Es ist allerdings nicht ganz einfach, ihnen mit den Augen zu folgen, weil es so viele sind. Versuchen wir es trotzdem, stellen wir fest, daß es Linien gibt, die alle Bausteine verbinden. Das ist der COMPUTERBUS (bus – Sammelschiene).

Stellen wir uns eine Autobuslinie vor, bei der die Haltestellen die elektronischen Bausteine sind. An diesen Haltestellen steigen natürlich keine Personen zu oder aus, beim Computerbus springen Impulse auf und ab. Wenn ihr auf eine Taste drückt, springen die acht Impulse (beim E sind es 0 1 1 0 0 1 0 1) gleichzeitig auf den DATENBUS und werden zum RAM transportiert. Dort steigen sie ab und werden festgehalten.

Zwischen Mikroprozessor, RAM und ROM gibt es noch einen anderen Bus, einen, der Adressen transportiert. Jeder Speicherplatz im Arbeitsspeicher hat eine Nummer. Es ist genau wie in einem Materiallager. Jedes Schubfach in den großen Regalen trägt ein Schild mit einer Nummer. Nach diesen Nummern werden die Waren einsortiert oder herausgenommen. Beim Speicher des Computers werden diese Nummern als Adressen bezeichnet. Sie geben dem Mikroprozessor an, wohin Befehle oder Daten gebracht werden sollen oder woher sie geholt werden können.

Diese Adressen werden wie Zahlen durch Impulse verschlüsselt. Bei einer Folge von acht Impulsen hatten wir 256 verschiedene Möglichkeiten zur Codierung. Das heißt aber nichts anderes, als daß wir nur 256 Speicherplätze von 0 bis 255 adressieren können. Ein einfacher Taschenrechner hat aber schon mehr Speicherplätze, und unser Computer braucht viele tausend davon. Die Adressen der Speicherplätze werden deshalb nicht nur mit acht, sondern mit sechzehn Impulsen verschlüsselt. Dadurch lassen sich 65 536 Speicherplätze

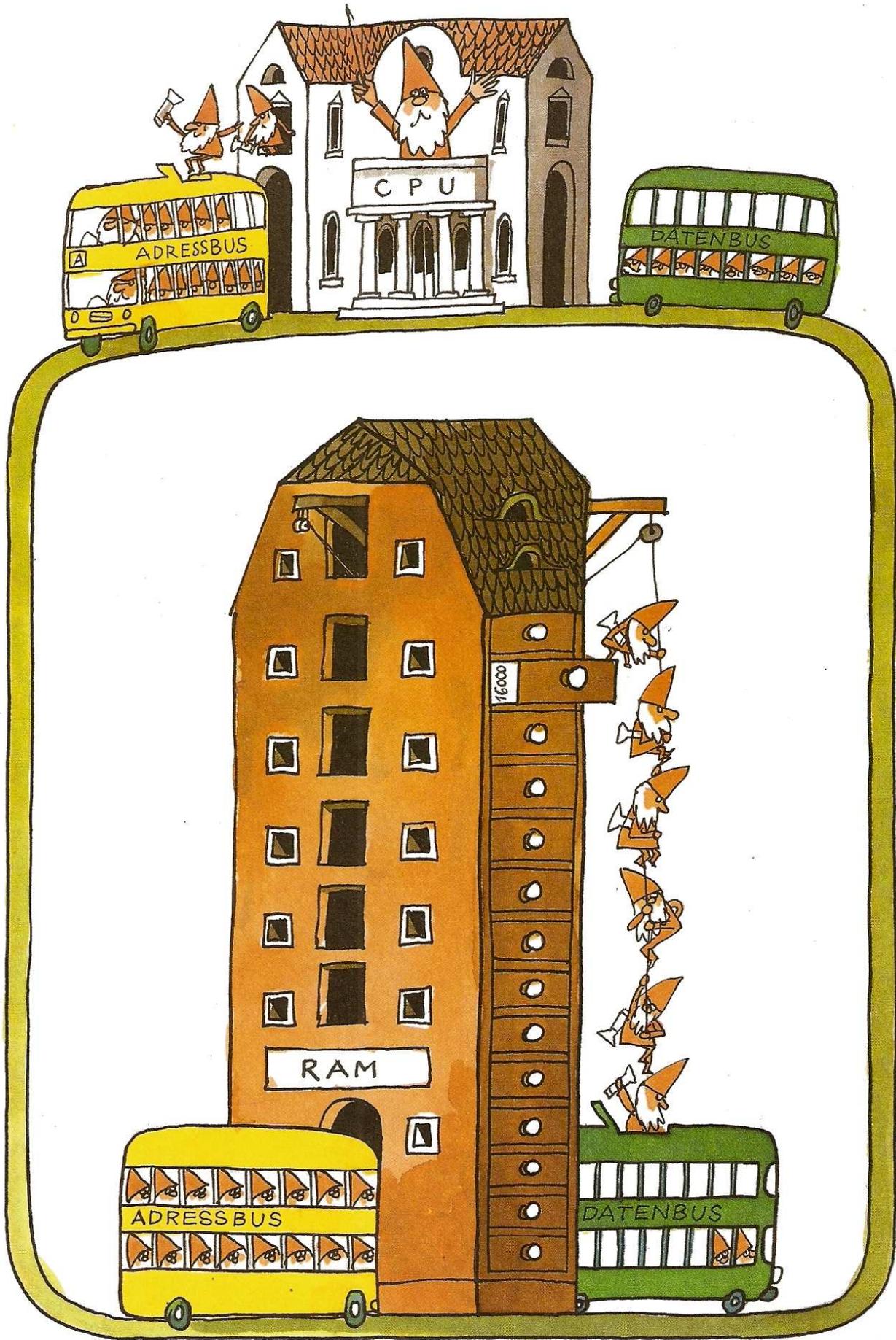
von 0 bis 65 535 adressieren. Und das ist schon eine ganze Menge.

Der Mikroprozessor soll jetzt einen Befehl abarbeiten, der sich in der Speicherzelle 16 000 befindet. Als erstes gibt er diese Adresse an. Sofort springen zweimal 8 Impulse auf den Adreßbus und signalisieren dem RAM, daß der Befehl in der Nummer 16 000 benötigt wird. Daraufhin springen die 8 Impulse, die in dieser Speicherzelle aufbewahrt wurden, auf den Datenbus und eilen in Blitzesschnelle zum Mikroprozessor. Das Besondere bei diesem Transport ist, daß der Inhalt in der Speicherzelle unverändert bleibt, so oft auch der Befehl, der Buchstabe oder die Zahl von der CPU abgerufen wird. Das ist wie im Schlaraffenland. Die Schüsseln und Gläser werden nie leer.

Außer dem Datenbus und dem Adreßbus haben wir noch einen dritten, den Steuerbus. Er verbindet den Mikroprozessor mit allen Bausteinen und Geräten, die an den Computer angeschlossen sind. Über ihn sorgt die CPU für die richtige Zusammenarbeit.

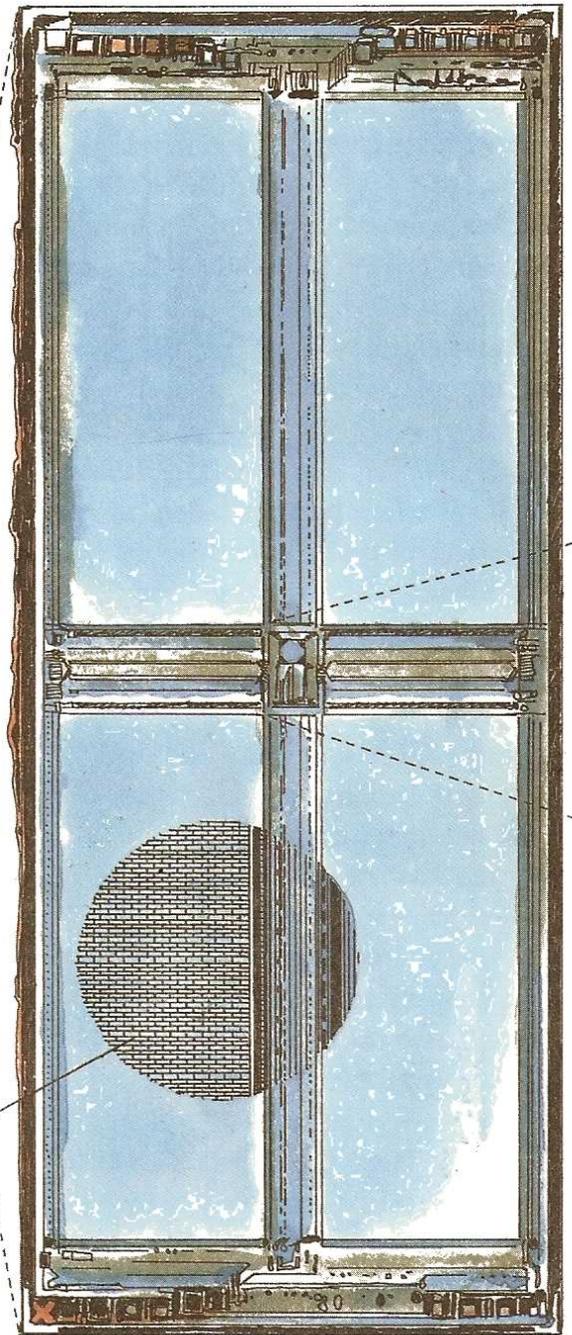
Damit nun keiner der Impulse oder Bausteine und keines der Geräte aus dem Arbeitsrhythmus kommt, hat der Mikroprozessor einen Taktgeber. Ihn benutzt er wie ein Dirigent als Taktstock, um das Spiel seines Orchesters zu dirigieren. Dazu schlägt er in einer Sekunde 2 Millionen Mal den Takt, seine Taktfrequenz ist also 2 MHz (Megahertz: Mega – Million, Hertz – Maßeinheit der Frequenz. Auf der Skala des Radios bei euch zu Hause könnt ihr das Zeichen MHz auch finden.). Wenn die Taktfrequenz erhöht wird, kann der Computer noch schneller arbeiten. Deshalb gibt es Computer mit Taktfrequenzen von 4, 8 oder noch mehr Megahertz.

Nun fragen wir uns natürlich, wie in so kleine Bausteine mehrere tausend und inzwischen sogar Millionen Speicherplätze hineinpassen. Dabei sind die Maikäfer nur die Hülle dieses Wunderwerks, das einfach nur CHIP (chip – Schnipsel) heißt. Diese Chips haben

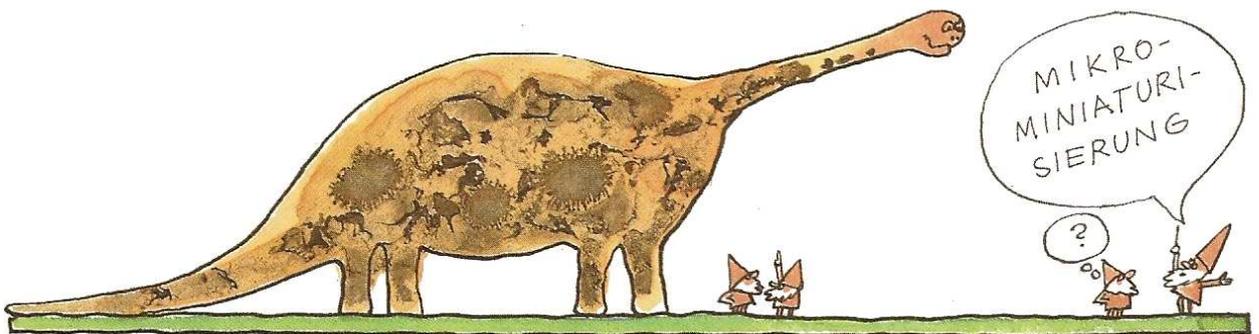




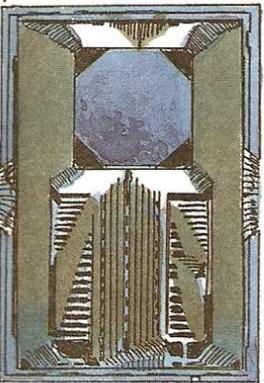
RAM-CHIP mit 272144 Speicherzellen
in 70-facher
Vergrößerung



in 14-facher Vergrößerung

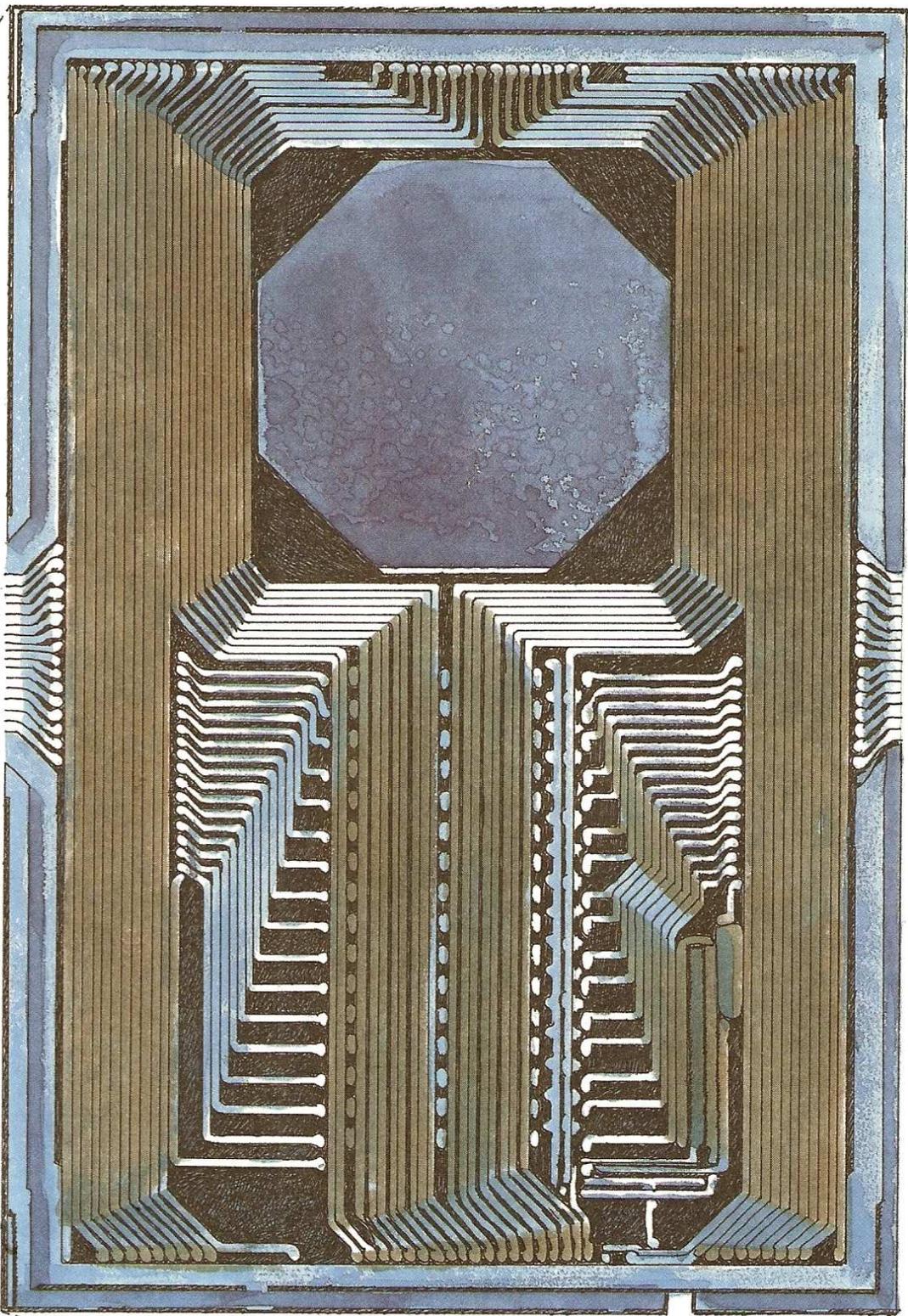


MITTE
DES
CHIPS



in 70-facher
Vergrößerung

MITTE
DES
CHIPS
MIT
ZENTRALEN
SCHALTUNGEN
in 300-facher
Vergrößerung



MIKROMINIATURISIERTER CHIP
in wirklicher Größe



KLEINST-
VER-
KLEINE-
RUNG



keinerlei Ähnlichkeit mit den Kartoffelchips, die ihr so gerne eßt. Der Chip des RAM-Bausteins ist kleiner als $1\text{ cm} \times 1\text{ cm}$, nicht einmal so groß wie euer Daumennagel. Auf ihm befinden sich Tausende von elektronischen Schaltkreisen. Solche gibt es auch im Radio oder im Fernsehgerät. Nur dort sind sie eben tausendmal größer. Was Chips leisten können, erfahrt ihr im nächsten Kapitel.

Wie rechnet der Computer?

Der Computer versteht, was wir ihm eintippen, obwohl er nur zwischen den Impulsen ein (1) und aus (0) unterscheiden kann. Das ist aber gar nicht so ungewöhnlich. Auf diese Weise verständigen wir uns nämlich auch bei anderen Gelegenheiten. Denkt nur daran, wie ihr eurem Freund klarmacht, daß ihr vor dem Haus steht und etwas wollt. Ein verabredetes Signal, z. B. vier Piffe – dreimal kurz, einmal lang –, genügt. Bei diesem Beispiel könnten wir vereinbaren, daß «kurz» die 1 und «lang» die 0 sein sollen.

Wenn ihr aufmerksam durch die Stadt geht, werdet ihr ein anderes Beispiel an vielen Straßenkreuzungen entdecken – die Verkehrsampel. Je nachdem, welche Lampe leuchtet, verhaltet ihr euch mit eurem Fahrrad anders. Ihr erhaltet über die Lampen der Ampel z. B. bei Grün die Information «Freie Fahrt». Zählen wir doch einmal nach, wieviel verschiedene Informationen die Verkehrsampel weitergibt. Es gibt also acht verschiedene Kombinationen der Lampen mit den Informationen:

- 0 Ampel außer Betrieb
- 1 Fahrzeuge halt
- 2 Achtung
- 3 Fahrzeuge zur Weiterfahrt bereitmachen
- 4 Freie Fahrt
- 5 Nicht genutzt
- 6 Vorsicht, Ampel schaltet in Kürze auf Gelb
- 7 Nicht genutzt.

22



Von den acht möglichen Informationen werden zwei aus Gründen der Verkehrssicherheit nicht genutzt.

Bei der Verkehrsampel werden durch das Ein oder Aus der Lampen 6 notwendige Informationen verschlüsselt. Wie sieht es aber aus, wenn viele tausend Informationen verschlüsselt werden sollen? Dabei wollen wir nicht jede Möglichkeit der Verschlüsselung ausprobieren oder abzählen. Es interessiert auch



nicht, welches der Hilfsmittel, ob Impuls, Lampe oder Pfiff, benutzt wird. Wichtig ist nur, daß zwei verschiedene Zustände (ein – aus, kurz – lang, 1 – 0) angegeben werden können. Die Anzahl der Impulse, Lampen oder Pfeife geben wir in BIT an (Abkürzung von binary digit – Binärzahl). Unser vereinbarter Pfiff enthält 4 Bit, die Lampen der Verkehrsampel ergeben 3 Bit und die 8 Impulse des Computers 8 Bit. Mit 3 Bit können $2 * 2 * 2 =$

$2^3 = 8$ Informationen und mit 8 Bit $2^8 = 256$ Informationen verschlüsselt werden. Mit 4 einzelnen Pfiffen, also 4 Bit, können demnach $2^4 = 16$ Informationen codiert werden. Zumindest die Hälfte der Schüler einer Klasse wären in der Lage, durch unterschiedliche Pfeife mitzuteilen, wer sich wo befindet: Peter: lang-lang-lang-lang (0000), Anna: lang-lang-lang-kurz (0001), Michael: lang-lang-kurz-lang (0010) usw. Ihr müßt nur aufpassen, daß andere Schüler nicht das gleiche System benutzen.

Inzwischen habt ihr längst bemerkt, wie wir auf die 65 536 verschiedenen Adressen bei den 16 Impulsen gekommen sind, denn $2^{16} = 65 536$. Die Möglichkeiten zur Verschlüsselung von Daten wachsen also mit der Zahl der Bits. Je mehr Bits zur Verfügung stehen, um so mehr Speicherplätze könnte der Mikroprozessor ansprechen.

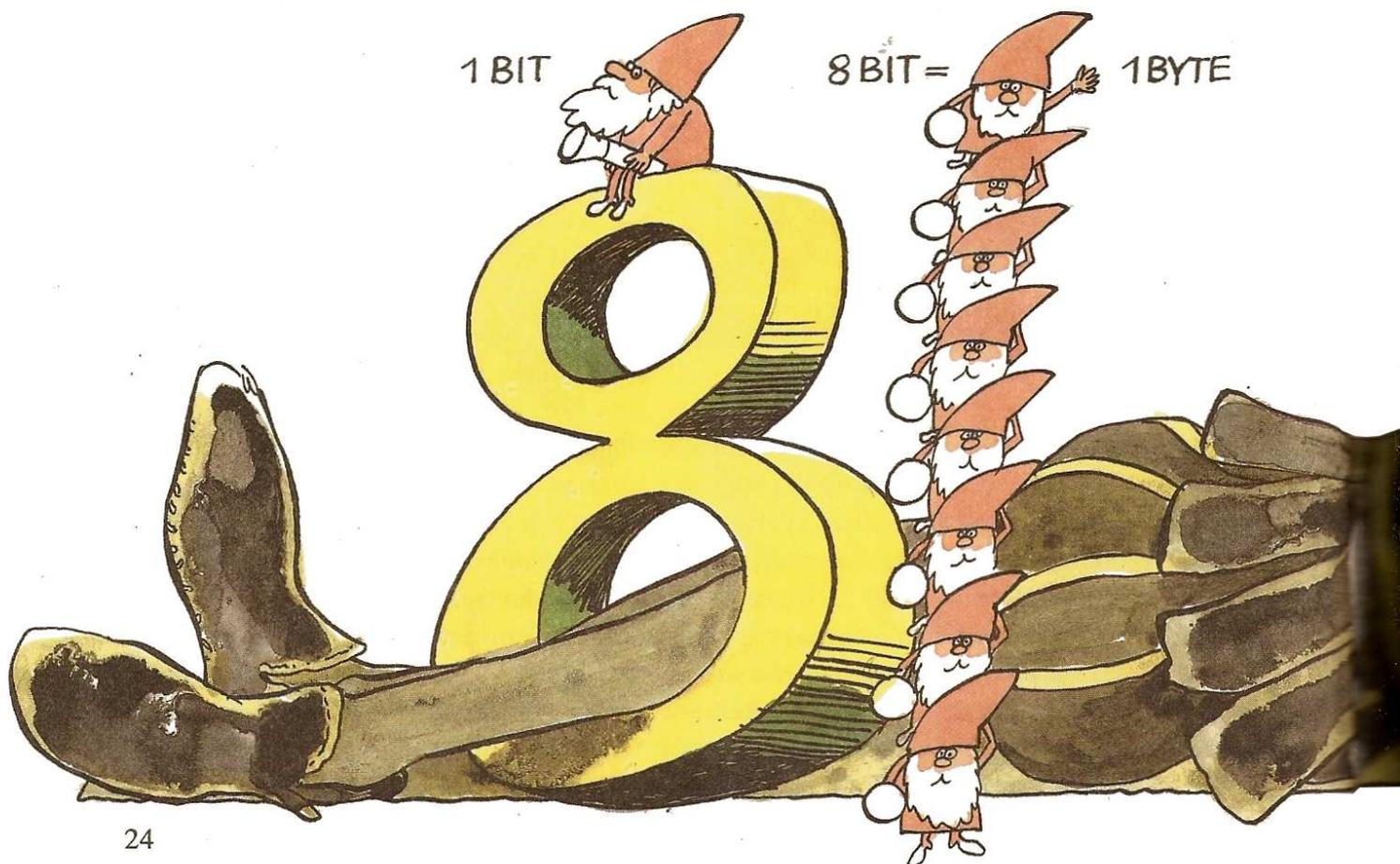
Die Bits verraten uns aber noch viel mehr. Ihre Anzahl gibt uns Auskunft darüber, wie schnell der Mikroprozessor ein Programm abarbeiten kann. Unser Mikroprozessor verarbeitet gleichzeitig 8 Impulse, weshalb er als 8-Bit-Prozessor bezeichnet wird. Je mehr Impulse gleichzeitig verarbeitet werden, um so schneller wird er sein. Ein 16-Bit-Prozessor ist beispielsweise 20mal schneller als der 8-Bit-Prozessor. Inzwischen arbeiten bereits Prozessoren mit 32 und mehr Bit. Der Aufbau dieser 32-Bit-Chips ist aber so kompliziert, daß sie erst in wenigen Ländern der Welt, darunter der DDR, hergestellt werden können. Für unsere Zwecke reichen die 8 Bit zunächst völlig aus. Wir wollen sogar noch ein paar davon weglassen, um besser zu verstehen, wie im Mikroprozessor gearbeitet wird. Dabei interessiert uns besonders, wie gerechnet wird und wie Buchstaben, Zahlen oder andere Zeichen miteinander verglichen werden. Diesen Teil des Mikroprozessors nennt man die ALU (Arithmetic and Logic Unit – Arithmetik- und-Logik-Einheit).

Daß man mit Zahlen rechnen kann, ist für euch natürlich ein alter Hut. Der Computer

rechnet aber mit Bits. Könnt ihr das auch? Wenn nicht, lest weiter, und schon könnt ihr es. Dazu schreiben wir noch einmal die Zahlen von 0 bis 15 in codierter Form auf. Links stehen die Dezimalzahlen und rechts die codierten Zahlen, die wir als Dualzahlen oder Binärzahlen bezeichnen. Das rührt daher, daß diese Zahlen nur durch zwei unterschiedliche Ziffern

dargestellt werden können (duo – zwei). Bei der Dezimalzahl lassen sich die Einer (10^0), Zehner (10^1), Hunderter (10^2) usw. immer durch zehn verschiedene Ziffern angeben. Anstelle der Einer, Zehner, Hunderter, Tausender im Dezimalsystem haben wir im Dualsystem die Einer (2^0), Zweier (2^1), Vierer (2^2), Achter (2^3), Sechzehner (2^4) usw.

Dezimalzahl	Dualzahl	Dezimalzahl	Dualzahl	Dezimalzahl	Dualzahl	Dezimalzahl	Dualzahl
0	0000	4	0100	8	1000	12	1100
1	0001	5	0101	9	1001	13	1101
2	0010	6	0110	10	1010	14	1110
3	0011	7	0111	11	1011	15	1111



Die Addition der Binärzahlen erfolgt nach den hier angegebenen Regeln:

$0 + 0 = 0$; $1 + 0 = 1$; $0 + 1 = 1$; $1 + 1 = 10$

$$\begin{array}{r} 5 \quad 0101 \\ + 4 \quad + 0100 \\ \hline 9 = 1001 \end{array}$$

$$\begin{array}{r} 3 \quad 0011 \\ + 12 \quad + 1100 \\ \hline 15 = 1111 \end{array}$$

Die Multiplikation von Binärzahlen ist besonders einfach:

$0 \times 0 = 0$; $0 \times 1 = 0$; $1 \times 0 = 0$; $1 \times 1 = 1$

Wenn der Computer zwei Dezimalzahlen addieren soll, werden sie bei der Eingabe über die Tastatur in Binärzahlen umgewandelt. Zur

Addition dieser Binärzahlen benötigt er drei verschiedene elektronische Schaltungen, die NICHT-, die UND- sowie die ODER-Schaltung, die zur ADDIER-Schaltung zusammengefaßt sind.

Jedes Bit hat seine eigene Addierschaltung auf dem Chip. Wenn 8 Bit gleichzeitig verarbeitet werden sollen, müssen acht von diesen Addierschaltungen auf dem Chip vorhanden sein. Jede Schaltung besteht wiederum aus mehreren Schaltkreisen. Es ist also ganz schön eng auf dem Chip. Damit alles Platz findet, müssen die Verbindungslinien in den Schaltkreisen hundertmal dünner als ein Haar sein.

Andere Schaltungen als die Addierschaltung hat der Computer nicht zum Rechnen. Es



wäre aber sehr traurig, wenn unser Computer nur addieren könnte. Keine Sorge, alle übrigen Rechenoperationen kann man mit Hilfe der Addition durchführen, denn $6 \times 3 = 3 + 3 + 3 + 3 + 3 + 3 + 3$ oder $6 - 3 = 6 + (-3)$. Das sieht zwar etwas umständlich aus, ist aber für den Computer kein Problem. Er arbeitet so schnell, daß für ihn ein paar Additionen mehr keine Rolle spielen. Oder kennt ihr jemanden, der wie unser Computer mehr als 1 Million Rechenoperationen pro Sekunde ausführen kann? Für die Multiplikation, Division oder andere mathematische Funktionen hat der Computer außerdem spezielle Programme im ROM gespeichert. Die werden, wenn ihr z. B. die Tasten * oder / drückt, abgearbeitet.

Bisher haben wir immer nur mit natürlichen Zahlen gerechnet. Es gibt aber noch andere Zahlenarten. Eine davon kennt ihr sicher vom Einkaufen, wenn ihr die Preise der Waren lest: 1,66 oder 25,97 M. Damit der Computer mit diesen ebenfalls rechnen kann, verschlüsselt er jede Zahl mit Hilfe von 5mal 8 Bit. Bei den Zahlen mit Komma (rationale Zahlen) wünscht der Computer bei der Eingabe allerdings statt des Kommas einen Punkt.

Vielleicht ist euch aufgefallen, daß die Zahl 8 sehr häufig auftaucht. Sie spielt wirklich eine besondere Rolle. Deshalb haben die 8 Bit einen eigenen Namen erhalten – BYTE. (Byte ist ein Kunstwort.) Wir brauchen für einen Buchstaben 1 Byte, für eine Adresse im Speicher 2 Byte und für eine Zahl 5 Byte. Auch die Anzahl der Speicherplätze des Computers wird gewöhnlich in Byte bzw. in Kilobyte (KByte) angegeben. Unser Computer hat eine Speicherkapazität von 64 KByte, also 65 536 Speicherplätze zu je 8 Bit.

Nanu, hier scheint irgend etwas nicht zu stimmen. Bei einem Kilo denken wir doch eigentlich an die Zahl 1 000 zur Umrechnung. Bei 64mal 1 000 kommen wir aber nicht auf die angegebene Zahl der Speicherplätze. Des Rätsels Lösung ist ganz einfach. Ein Computer-Kilo wird im Binärsystem ausgerechnet

und ist deshalb $2^{10} = 1 024$ und nicht 1 000. Es ist also alles O. K., wie der Computer sagt.

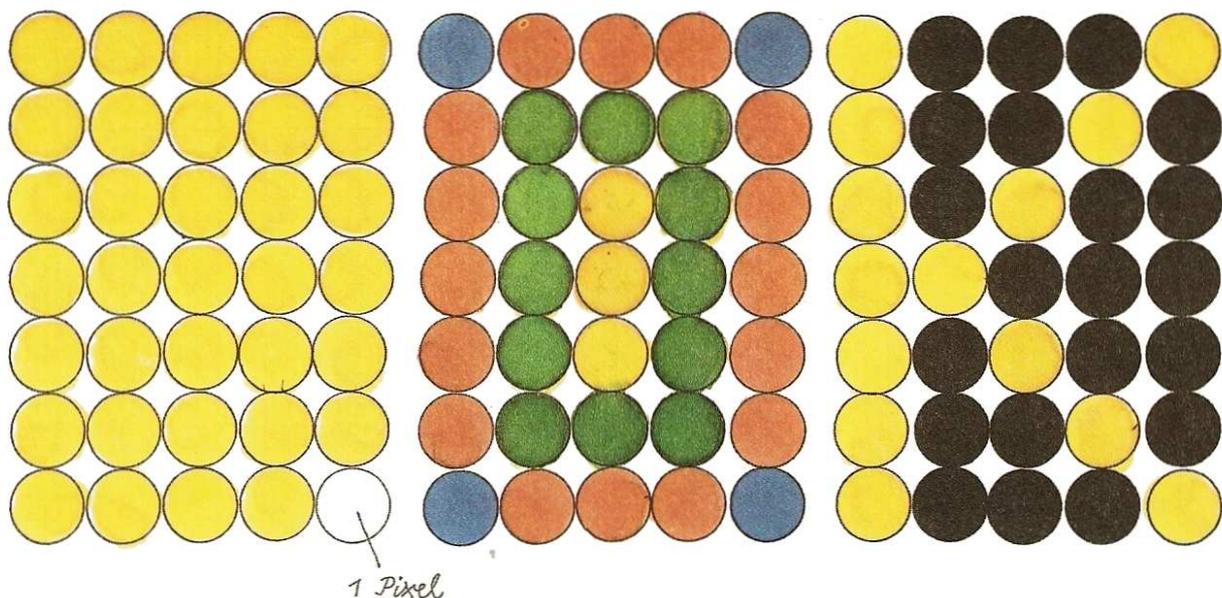
Der Bildschirm bringt es an den Tag

Hat der Computer keinen eigenen Bildschirm, kann man ihn an ein Fernsehgerät anschließen. Dann muß man sich entscheiden, ob man das Fernsehprogramm sehen will oder mit dem Computer vielleicht selbst Bilder auf den Bildschirm bringen möchte. Außerdem soll auch auf dem Bildschirm zu lesen sein, was der Computer für uns herausgefunden hat.

Wißt ihr eigentlich, daß der Bildschirm eines Fernsehgerätes aus rund 150 000 kleinen, länglichen Pünktchen zusammengesetzt ist, die zudem auch noch in unterschiedlichen Farben leuchten können? Nein? Dann geht einmal dicht heran. Sie sind deutlich zu sehen. Diese Pünktchen werden PIXEL genannt. Pixel ist eine Abkürzung aus picture element – Bildelement. Es weist darauf hin, daß jedes Bild auf dem Fernsehgerät aus einzelnen Elementen zusammengesetzt ist. In den Zeitungen sind die Bilder ebenfalls aus lauter Punkten zusammengesetzt. Selbst ohne Lupe ist das zu erkennen.

Unser Computer muß also die Pixel des Bildschirms dazu bringen, daß sie in der von uns gewünschten Farbe leuchten. Dazu muß er ihnen mehrmals in der Sekunde Impulse schicken, denn die Pixel leuchten nur für eine sehr kurze Zeit. Das hat seinen Sinn. Würde nämlich jeder Punkt fortwährend leuchten, könnten wir auf dem Bildschirm keine bewegten Bilder sehen. Es würde alles verschwimmen – wie beim Fotografieren, wenn wir zwei oder mehr Bilder übereinanderknipsen.

Für jeden Pixel ist die Information, ob er leuchten soll oder nicht, im RAM gespeichert. Bei einem Schwarzweißbild genügt dafür ein Bit je Pixel. Bei farbiger Darstellung benötigt



der Computer entsprechend der Farbenzahl mehr Bits und damit mehr Platz im RAM. Dieser reservierte Speicherplatz (der BILDWIEDERHOLSPEICHER) steht uns dann für andere Daten nicht mehr zur Verfügung. Wenn der Computer nur insgesamt 64 KByte Speicher hat, wovon 48 KByte zum RAM gehören, können wir demnach doch nicht ganz so großzügig mit den Bytes verfahren, wie das am Anfang aussah.

Wollen wir nun, daß der Computer Zahlen oder Buchstaben schreibt, brauchen wir uns nicht den Kopf zu zerbrechen. Mit dem Befehl PRINT (print – drucken) erzeugt er die gewünschten Zeichen. Dazu dient ihm ein Raster, welches stark vergrößert wie Rechenkästchenpapier aussieht. Deutlich sind 7 Zeilen und 5 Spalten, also 35 Kästchen, zu erkennen. Es gibt aber auch Computer, bei denen das Raster aus $8 \text{ mal } 8 = 64$ Kästchen besteht. Das Prinzip für die Darstellung der Zeichen ist aber immer das gleiche. Schaltet der Computer bestimmte Kästchen ein (1), während die anderen aus (0) bleiben, kann er, wie ihr seht, jedes beliebige Zeichen der Tastatur darstellen: A–Z, 0–9, / + & .

Den Text, den wir drucken wollen, müssen wir mit dem Zeichen " einschließen. In unse-

rem Programm ZUG könnten wir nun den Druckbefehl ändern:

```
40 PRINT "Geschwindigkeit =", GESCHWINDIGKEIT; "km/h", so daß auf dem Bildschirm erscheint:
```

```
Geschwindigkeit = 120 km/h.
```

Genausogut könnten wir uns etwas anderes ausdenken, z. B.

```
10 PRINT "Peter liebt Anna!",  
und auf dem Bildschirm lest ihr:
```

```
Peter liebt Anna!
```

Da wir dem Computer nicht vorgegeben haben, an welcher Stelle des Bildschirms er mit dem Drucken beginnen soll, fängt er automatisch links oben an. Wenn wir eine leere Seite beschreiben, tun wir das ja normalerweise auch dort. Oft soll aber der Text an einer bestimmten Stelle des Bildschirms erscheinen. Dann hilft uns wiederum ein Raster. Je nachdem, welchen Computertyp ihr verwendet, kann es etwas anderes sein. Deshalb geben wir euch hier nur ein Beispiel von mehreren Möglichkeiten an. Wir nehmen Rechenkästchenpapier und numerieren die Zeilen von links oben nach links unten von 0 bis 21. Die Spalten werden von links oben nach rechts oben von 0 bis 31 numeriert. Auf diese Weise er-



Spalten
Zeilen

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
0																																	
1																																	
2																																	
3																																	
4																																	
5																																	
6																																	
7																																	
8																																	
9																																	
10																																	
11																																	
12																																	
13																																	
14																																	
15																																	
16																																	
17																																	
18																																	
19																																	
20																																	
21																																	

haltet ihr ein Raster mit 704 Kästchen oder Feldern. Für jedes Zeichen wird ein Kästchen benötigt. Eine Zeile auf dem Bildschirm hat demnach Platz für 32 Zeichen. Die Zeilen 22 und 23 des Bildschirms behält sich der Computer für die Eingabe von Kommandos oder Fehlermeldungen vor. Jetzt könnt ihr jedes Feld auf dem Bildschirm durch die Angabe von 2 Zahlen finden. Erst geben wir die Zeilennummer an, dann die der Spalte.

Wir wollen nun unseren Text in die Mitte des Bildschirms setzen, d. h. auf Zeile 11. Der Text mit Peter und Anna hat einschließlich der Zwischenräume 18 Zeichen. Deshalb fangen wir beim Druck bei Spalte 7 zu schreiben an. Es bleiben also 7 Kästchen frei. 18 Kästchen brauchen wir für den Text. Dahinter bleiben wiederum 7 Kästchen frei. Das ergibt genau 32 Kästchen einer Zeile. Um das zu erreichen, wird der Druckbefehl durch eine Ortsangabe erweitert. Er heißt jetzt PRINT AT (drucke an der Stelle), und seine vollständige Form wäre für unser Beispiel:

10 PRINT AT 11, 7; "Peter liebt Anna!"
Soll nun noch ein Bild von Anna und Peter auf dem Bildschirm erscheinen, kommen wir mit unserem groben Raster nicht mehr aus. Wir müssen dafür die einzelnen Pixel verwenden. Wie man so etwas machen kann, werdet ihr in einem der nächsten Abschnitte erfahren. Inzwischen könnt ihr aber das Programm ZUG

etwas verschönern. Wie wäre es mit einer Überschrift? Außerdem könnte ein Antwortsatz nicht schaden.

```

5 PRINT AT 1, 9; "Programm ZUG"
10 INPUT ZEIT
20 INPUT WEG
30 LET GESCHWINDIGKEIT =
   WEG/ZEIT
40 PRINT AT 5, 2; "Die Geschwindigkeit
   des D-Zuges betrug"; GESCHWIN-
   DIGKEIT; " km/h".
50 GOTO 10

```

Das Programm läuft

Kennt ihr das Spiel Zahlenraten? Auch wer es nicht kennt, versteht bald, worum es geht. Ein Spieler denkt sich eine Zahl zwischen 1 und 100 aus und schreibt diese auf einen Zettel. Er könnte ja sonst schummeln. Die anderen Spieler versuchen, diese Zahl zu erraten. Der einzelne Spieler antwortet, solange die richtige Zahl nicht gefunden wurde, nur mit dem Hinweis, daß die genannte Zahl zu groß oder zu klein ist. Wer die wenigsten Versuche benötigt, um die richtige Zahl zu erraten, ist der Sieger. Allein kann man dieses Spiel natürlich nicht spielen. Aber wie wäre es, wenn sich der Computer die Zahl ausdenkt, die ihr erraten sollt? Dazu schreiben wir das Programm



”ZAHLENRATEN”. Damit wir uns in dem Programm zurechtfinden, benutzen wir einen Programmablaufplan (Flußdiagramm). Hierin steht alles, was in dem Programm passieren soll. Die Pfeile zeigen die Richtung an, in die wir uns im Programm zu bewegen haben.

Wir beginnen damit, daß sich der Computer meldet und fragt, wie oft ihr raten wollt. Dann rechnet er sich die Zahl aus, die ihr erraten sollt. Anschließend kontrolliert er, ob ihr schon öfter geraten habt, als ihr vereinbart hattet. Wenn das der Fall sein sollte (Richtung ja), gibt der Computer seine Zahl bekannt und fragt, ob ihr eine neue Zahl erraten wollt.

Habt ihr noch Versuche (Richtung nein), fragt er euch nach eurer Zahl. Diese wird mit der Zahl des Computers verglichen. Habt ihr richtig geraten (Richtung ja), stellt der Computer fest, wie viele Versuche ihr gebraucht habt, und druckt eine entsprechende Bewertung. Anschließend fragt er, ob ihr eine neue Zahl raten wollt.

War eure Zahl nicht richtig (Richtung nein), dann stellt der Computer fest, ob eure Zahl größer oder kleiner als seine Zahl war, und teilt euch das mit. Anschließend kehrt er zu der Stelle im Programm zurück, wo er fest-

stellt, ob ihr noch Rateversuche habt. Ist das der Fall, dann fragt er euch wieder nach eurer neuen Zahl.

Den Druck- und den Eingabebefehl kennt ihr bereits aus dem Programm ZUG. Bei dem Eingabebefehl: 30 INPUT ”Wie oft willst du raten?”, N, druckt der Computer: Wie oft willst du raten? Der blinkende Cursor zeigt euch, daß er auf eine Eingabe (INPUT) einer Zahl wartet. Tippt ihr jetzt zum Beispiel eine 10 ein, dann erhält N den Wert 10, und ihr habt 10 Rateversuche.

Für das Zahlenraten benötigen wir aber noch ein paar andere Befehle. Da wäre zunächst der Befehl CLS (clear screen – lösche den Bildschirm). Es empfiehlt sich zu Beginn des Programms und manchmal auch, wenn auf dem Bildschirm zuviel Text steht, auf ihm sämtliche Zeichen zu löschen.

Mit Hilfe von LET (let – lassen) vereinbart der Computer, welchen Zahlenwert oder Inhalt ein Buchstabe erhalten soll. Einige Beispiele sollen euch das zeigen.

LET A = 3; der Buchstabe A erhält den Wert 3

LET B = 3 + 6; der Buchstabe B erhält den Wert 9

LET A = B; A erhält den Zahlenwert von B. B bleibt unverändert.

LET A = B + 7; A erhält den Zahlenwert der Summe B + 7.

LET A = A * A; A erhält den Zahlenwert von A * A. Der alte Zahlenwert von A wird durch den neuen Wert A * A überschrieben und ist damit verloren.

Wie ihr in dem Programmablaufplan gesehen habt, läuft das Programm nicht geradlinig ab. Es gibt auch Verzweigungen. In welcher Richtung das Programm an diesen Punkten fortgesetzt wird, entscheidet der Computer mit Hilfe des Befehles IF (if – falls, wenn) und THEN (then – dann).

Schreiben wir z. B. IF Y = X THEN 200, heißt das: falls der Wert von Y gleich dem von X ist, dann setze das Programm beim Befehl mit der Nummer 200 fort (Richtung ja). Ist X dagegen ungleich Y, wird der nächste Befehl in der Programmfolge abgearbeitet (Richtung nein).

Soll ein bestimmter Programmteil übersprungen werden, benutzen wir den Befehl

GOTO (go to – gehe zu) und geben die Nummer der Programmzeile an, bei der das Programm fortgesetzt werden soll. Beispiel: 140 GOTO 50

Ein Befehl, der in fast jedem Spielprogramm benötigt wird, ist der Befehl zur Erzeugung einer Zahl, deren Wert nicht vorausgesagt werden kann. Solche Zahlen nennt man Zufallszahlen. Deshalb heißt auch der Befehl RND, was von random – zufällig kommt. Die meisten Mikrorechner haben im ROM ein kleines Programm, mit dessen Hilfe Zahlen zwischen 0 und 0.99999999 zufällig erzeugt werden, z. B. die Zahl 0.57869143. Will man aus dieser Zahl eine größere Zufallszahl machen, braucht man sie nur mit 10 oder 100 zu multiplizieren: 5.7869143 oder 57.869143.

Oft interessiert uns nur der Wert einer Zahl vor dem Dezimalpunkt. Mit Hilfe der Anweisung LET X = INT X (integer – ganze Zahl) wird aus X = 57.869143 die Zahl X = 57.

Wir numerieren die Befehle wieder in Zehnerschritten, damit ihr nachträglich noch Befehle einfügen könnt, ohne das Programm umschreiben zu müssen. Los geht's!

10 REM Zahlenraten

Den Befehl REM (remark – bemerken) benutzen wir, um uns im Programm besser zurechtzufinden. Hier merken wir uns, um welches Programm es sich handelt. Wenn das Programm sehr lang ist, muß man nicht erst viele Seiten lesen, um herauszufinden, was programmiert wurde. Der Computer überliest diesen Befehl. Das Programm wird nicht beeinflusst.

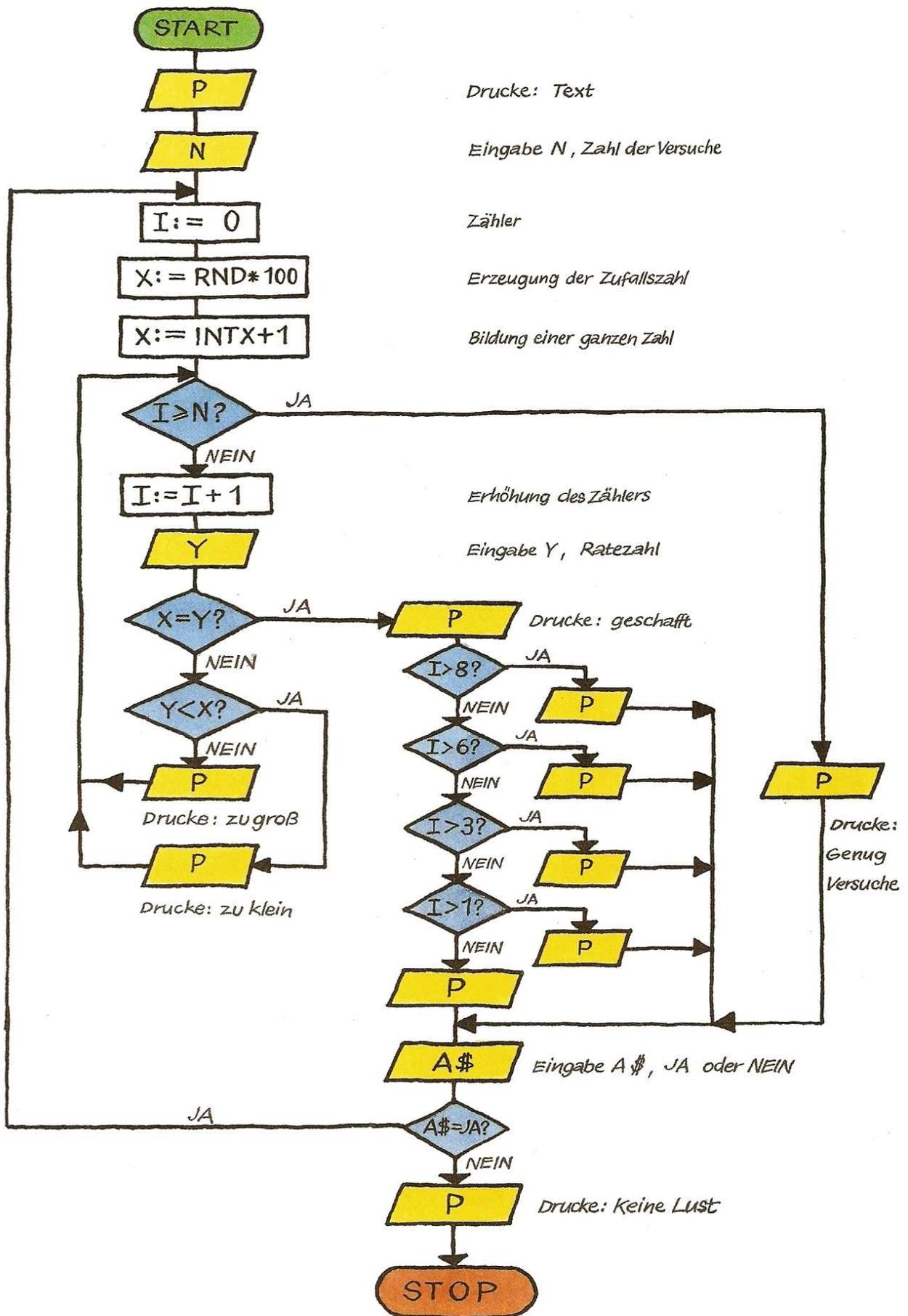
20 CLS

Der Bildschirm wird gelöscht.

30 PRINT "Ich habe mir eine Zahl zwischen 1 und 100 ausgedacht!"

40 INPUT "Wie oft willst du raten?", N

Der Cursor zeigt euch, daß er auf die Eingabe (INPUT) einer Zahl wartet. Tippt ihr jetzt zum Beispiel eine 10 ein, dann hat N den Wert 10. Ihr dürft dann zehnmal raten.



```

10 REM Zahlenraten
20 CLS
30 PRINT "Ich habe mir eine Zahl
    zwischen 1 und 100 ausgedacht
    ."
40 INPUT "Wie oft willst du raten?" N
50 LET I=0
60 CLS
70 REM Zufallszahl
80 LET X=RND*100
90 LET X=INT X+1
100 IF I>=N THEN GO TO 330
110 LET I=I+1
120 INPUT "Welche Zahl ist es?"
    Y
130 IF X=Y THEN GO TO 200
140 IF Y<X THEN GO TO 170
150 PRINT Y;" ist zu gross!"
160 GO TO 100
170 PRINT Y;" ist zu klein!"
180 GO TO 100

```



50 LET I = 0

Wir brauchen in unserem Programm I als Zähler. Zu Beginn muß er Null sein, denn nach zehn Versuchen steht er auf zehn, wenn ihr die Zahl nicht vorher erraten habt. Dann hat der Computer gewonnen.

60 CLS

Siehe Zeile 20.

70 REM Zufallszahl

Der Computer «denkt» sich eine Zahl aus.

80 LET X = RND * 100

Es wird eine Zufallszahl berechnet. Ihr Wert liegt zwischen 0 und 99.999999.

90 LET X = INT X + 1

Aus X wird eine ganze Zahl gebildet. Damit die Null ausscheidet, wird eine 1 addiert. Wir nehmen als Beispiel an, daß X = 58 ist.

100 IF I >= N THEN 330

Beim ersten Durchlauf des Programms heißt der Befehl 100: Wenn (IF) N größer oder gleich 10 ist, dann (THEN) gehe zum Befehl 330. Null ist kleiner als 10. Deshalb betrachtet der Computer die Bedingung als falsch und geht nicht zum Befehl 330, sondern zum Befehl 110.

Wenn also der Zähler I größer oder gleich der Höchstzahl der Rateversuche N ist, dann wird das Programm vom Befehl mit der Nummer 330 ab fortgesetzt. Wenn die Vereinbarung nicht wahr ist, wie das beim ersten Versuch der Fall ist, wird der nächste Befehl abgearbeitet.

110 LET I = I + 1

Der Wert des Zählers wird um 1 erhöht, um die Versuche mitzuzählen. I erhält also jetzt den Wert 1.

120 INPUT "Welche Zahl ist es?", Y

Es wird nach der zu erratenden Zahl gefragt. Wir nehmen an, ihr tippt eine 60 ein. Dann erhält Y den Wert 60.

130 IF X = Y THEN 200

Jetzt überprüft der Computer, ob seine Zahl X mit eurer Zahl Y übereinstimmt. Der Befehl lautet demnach: Falls X gleich Y ist, dann gehe zum Befehl 200. Wenn X ungleich Y ist, wird der nächste Befehl des Programms abgearbeitet. Da 58 ungleich 60 ist, geht der Computer zum nächsten Befehl.

140 IF Y < X THEN 170

Es wird nun geprüft, ob eure Zahl kleiner ist als die des Computers. Wenn das der Fall ist, dann wird das Programm ab Befehl 170 fortgesetzt. Bei unserem Beispiel stellt der Computer fest, daß die Frage mit Nein beantwortet werden muß. Eure Zahl ist größer als die des Computers. Deshalb geht er weiter zum nächsten Befehl.

150 PRINT Y; "ist zu gross!"

Auf dem Bildschirm erscheint: 60 ist zu gross!

160 GOTO 100

Weil nicht richtig geraten wurde, wird ein neuer Versuch begonnen. Dazu muß der Computer zum Befehl 100 gehen.

170 PRINT Y; "ist zu klein!"

Hierher gelangen wir vom Befehl 140, wenn eure Zahl kleiner als die zu erratende Zahl ist.

180 GOTO 100

Neuer Versuch ab Befehl 100.

200 PRINT "Du hast es geschafft!"
"Meine Zahl war"; X; "."

Zu diesem Befehl gelangt ihr nur, wenn ihr die Zahl erratet, bevor ihr mehr als 10 Versuche gemacht habt. Der PRINT-Befehl unterscheidet sich etwas von den bisherigen PRINT-Befehlen. Das Zeichen ' benutzt man, wenn nach dem Drucken eines Textstückes eine neue Zeile begonnen werden soll. Das Semikolon (;) wird verwendet, wenn die nachfolgenden Zeichen in der gleichen Zeile auf dem Bildschirm stehen sollen wie der vorher geschriebene Text. In unserem Beispiel würde gedruckt: Du hast es geschafft! Meine Zahl war 58.

205 REM Bewertung

In diesem Teil des Programms wird festgestellt, wie gut ihr geraten habt.

210 IF I > 8 THEN 260

Es wird geprüft, wieviel Versuche ihr gemacht habt. Falls die Anzahl der Versuche I größer als 8 war, dann gehe zum Befehl 260.

220 IF I > 6 THEN 270

230 IF I > 3 THEN 280

240 IF I > 1 THEN 290

250 PRINT "Du bist super!"
"Ich gratuliere!"; GOTO 300

Wenn der Computer diesen Befehl erreicht, dann habt ihr die Zahl auf Anhieb erraten. Will man mehrere Befehle in eine Zeile schreiben, trennt man sie durch einen Doppelpunkt.

260 PRINT "Das war aber knapp.""
Immerhin ";I; " Versuche!"; GOTO 300

Wurden mehr als 8 Versuche benötigt, gelangt der Computer hierher und springt dann zum Befehl 300.

```
270 PRINT "Na ja, ";I;" mal geraten."  
""Warst auch schon besser!": GOTO 300
```

```
280 PRINT "Du bist schon ganz gut."  
""Aber trotzdem ";I; "Versuche!":  
GOTO 300
```

```
290 PRINT "Die Meisternorm hast  
du erreicht."""Mein Glückwunsch!  
""Sind vielleicht auch weniger  
als 2 Versuche drin?"
```

```
300 INPUT "Willst du ein neues Spiel?  
Antworte mit JA oder NEIN!", A$
```

Der Computer erwartet wieder eine Eingabe. Durch das Zeichen \$ hinter dem Buchstaben betrachtet er alles, was ihr eintippt, als Buchstaben. Dabei wird auch zwischen großen und kleinen Buchstaben unterschieden. Der Cursor blinkt in solch einem Fall zwischen zwei "-Zeichen.

```
310 IF A$ = "JA" THEN 20
```

Wenn ihr mit JA geantwortet habt, ist die Anweisung 310 wahr. Der Computer springt im Programm zum Befehl 20. Bei jeder anderen Eingabe geht er weiter im Programm.

```
315 CLS
```

siehe Zeile 20

```
320 PRINT "Wohl keine Lust mehr?"  
""Na dann mach's gut!": GOTO 350
```

```
330 PRINT "Das waren genug Ver-  
suche!""Ich verrate dir meine  
Zahl."""Es war die ";X: GOTO 300
```

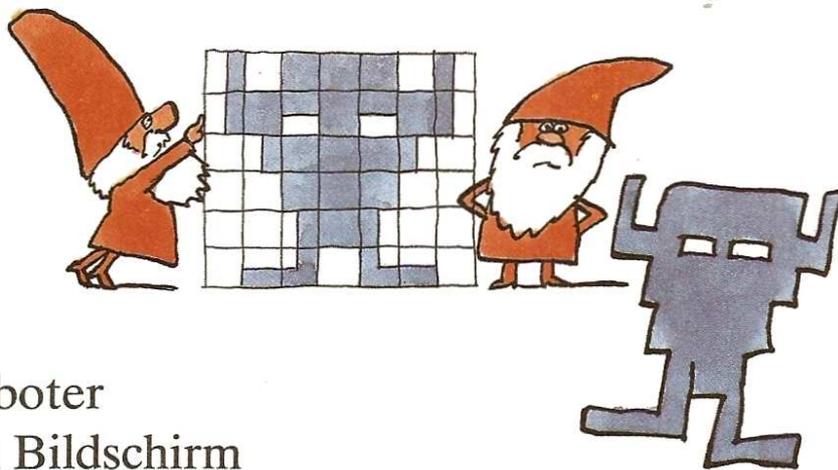
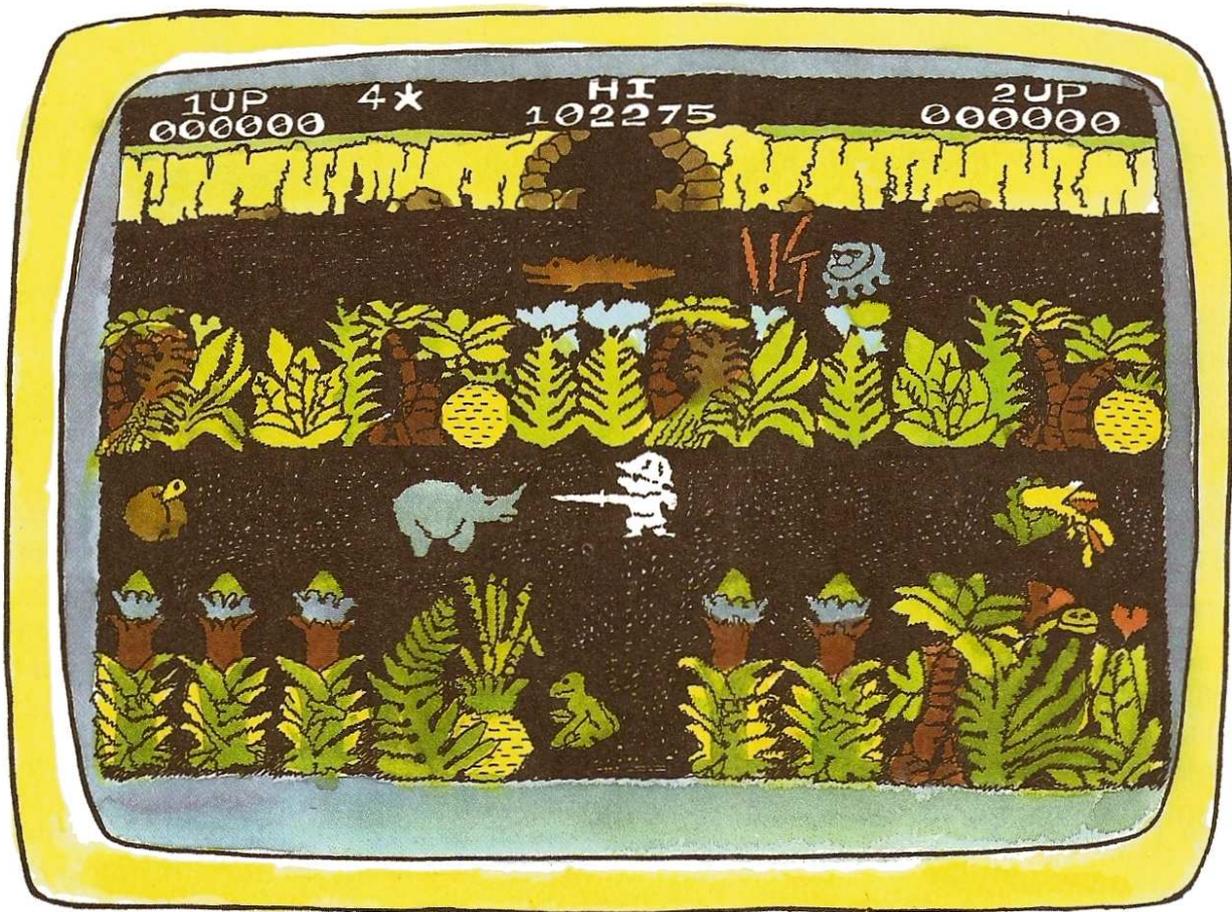
```
350 STOP
```

Der STOP-Befehl beendet die Abarbeitung des Programms.

```
360 END
```

Der Computer erfährt über diesen Befehl, daß das Ende des Programms erreicht wurde.

Nach dem Eintippen wird das Programm durch RUN gestartet. Aber das wißt ihr ja bereits. Also viel Spaß beim Raten.

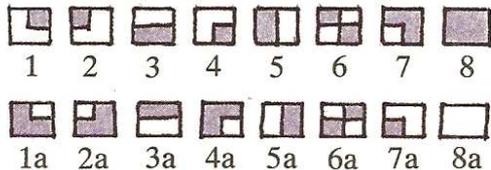


Der Roboter auf dem Bildschirm

In den letzten Jahren sind Computerspiele programmiert worden, bei denen auf dem Bildschirm Figuren durch Urwälder laufen, mit dem Flugzeug über Seen, Flüsse, Städte und Wolken fliegen oder mit einem Unterseeboot bizarren Landschaften und Fischen begegnen. Programme dieser Art kann man nicht

von heute auf morgen herstellen. Ein guter Programmierer arbeitet daran ungefähr ein Jahr. Das Prinzip vieler Spiele ist dabei oft relativ einfach. Der große Programmieraufwand ergibt sich aus den Bildern. Aber erst durch die wechselnden Bilder werden die Spiele so eindrucksvoll.

Selbst wenn nur ganz einfache Bilder entstehen sollen, müssen wir schon eine Menge tun. Bei manchen Computern finden wir auf der Tastatur bereits Zeichen, aus denen sich Figuren zusammensetzen lassen. Mit Hilfe einer speziellen Taste GRAPHIC können wir diese Bildelemente aufrufen. Ihr seht hier acht dieser Elemente. Zusätzlich erhalten wir noch einmal acht Elemente, wenn die hellen und dunklen Teile des Feldes vertauscht werden.



Die Größe der Kästchen (ein Kästchen besteht aus 5×7 oder 8×8 Pixeln) entspricht einem Feld auf eurem Raster mit den 22 Zeilen und 32 Spalten. Wollt ihr einmal probieren, ob ihr einen Roboter auf den Bildschirm zaubern könnt? Mit dem folgenden Programm müßte es gelingen. Legt aber zuerst das Raster neben das Buch und zeichnet die einzelnen Bildelemente an den im Programm angegebenen Stellen ein. So könnt ihr vergleichen, ob dann das Programm richtig funktioniert. Achtet aber darauf, welches Raster von eurem Computer benutzt wird. Wir verwenden hier wieder 5×7 Pixel und das 32-Spalten-mal-22-Zeilen-Raster. (Spalten von 0–31 und Zeilen von 0–21)

```
5 REM Roboter
10 CLS
20 PRINT AT 3,13;"
```

In Zeile 3 wird ab Spalte 13 mit dem Drucken begonnen. Wir benutzen die Elemente 5a, 8a, 8, 8, 8, 8a, 5. Bei den Elementen mit einem a sind hell und dunkel vertauscht. Auf diese Weise fahren wir fort und setzen den Roboter zusammen.

```
30 PRINT AT 4,13;"
    Elemente 5a, 8a, 8, 8, 8, 8a, 5.
40 PRINT AT 5,13;"
    Elemente 5a, 8, 3a, 8, 3a, 8, 5.
```

```
50 PRINT AT 6,14;"
    Elemente 5a, 8, 8, 8, 5.
60 PRINT AT 7,15;"
    Elemente 8, 8, 8.
70 PRINT AT 8,15;"
    Elemente 5a, 8, 5.
80 PRINT AT 9,14;"
    Elemente 3, 2a, 8a, 1a, 3.
90 STOP
```

Auf dem Papier und auf dem Bildschirm müßte jetzt ein Roboter stehen. Soll er auf dem Bildschirm einmal hier und dann wieder dort auftauchen, lassen wir den Computer die Zeile und Spalte, wo er mit dem Drucken beginnt, selbst auswählen. Vielleicht probiert ihr es gemeinsam im Computerzirkel. Wie wir den Roboter programmieren würden, haben wir euch aufgeschrieben.

```
5 REM Roboter Mobil
10 CLS
15 LET I = INT (RND * 10) + 2
16 LET J = INT (RND * 20) + 5
20 PRINT AT I, J;"
30 PRINT AT I + 1, J;"
40 PRINT AT I + 2, J;"
50 PRINT AT I + 3, J + 1;"
60 PRINT AT I + 4, J + 2;"
70 PRINT AT I + 5, J + 2;"
80 PRINT AT I + 6, J + 1;"
90 LET L = 1
100 IF L > 31 THEN 10
110 PRINT AT 21, L;"
120 LET I = L + 1: GOTO 100
130 STOP
```

Startet ihr jetzt das Programm, erscheint der Roboter nach jedem Programmdurchlauf an einer anderen Stelle des Bildschirms. Vielleicht wollt ihr ihm noch Antennen an den Kopf setzen, oder ihr laßt ihn Fußball spielen. Bis jetzt kann er das allerdings nur mit einem eckigen Ball. Das gefiele euch sicher nicht. Dann machen wir ihn eben rund.

Dazu benötigt ihr aber einen neuen Befehl und ein neues Raster für die Zeilen und Spalten auf dem Bildschirm. Wir wollen dort nämlich jeden einzelnen Pixel finden. Am besten, ihr nehmt einen Bogen Millimeterpapier, denn jedes der Kästchen auf dem alten Raster muß in 64 kleine Felder unterteilt werden. Es besteht aus genau $8 \text{ mal } 8 = 64$ Pixeln. Auf diese Weise erhalten wir 176 neue Zeilen und 256 Spalten. Wenn euer Computer einen anderen Bildschirm hat, lest ihr im Handbuch nach, wie viele Zeilen und Spalten er hat. Wenn ihr die Zeilen von 0 bis 175 und die Spalten von 0 bis 255 durchnummeriert, seht ihr es schon. Auch dieses Mal wird die Position der kleinen Felder durch zwei Zahlen bestimmt. Damit der Computer unterscheiden kann, ob ihr ein großes Feld oder nur ein Pixel

drucken wollt, verwenden wir für die Pixel einen anderen Druckbefehl – PLOT. Das heißt soviel wie Fleck. Wir müssen allerdings beachten, daß beim PLOT-Befehl erst die Nummer X der Spalte und dann die Zahl Y der Zeile angegeben wird. Jetzt könntet ihr den Ball aufblasen. Dazu setzen wir das Programm Roboter beim Befehl 90 anders fort, nehmen das Bildelement 8 und ergänzen die fehlenden Punkte, bis eine Kugel entsteht.

Dieser Programmteil läßt sich natürlich noch bequemer programmieren. Wir haben euch die einfachste Form aufgeschrieben, damit ihr auf dem Papier überprüfen könnt, ob am Ende auch wirklich ein Kreis zu sehen ist.

Ihr habt bestimmt schon viele Ideen, was man alles auf dem Bildschirm darstellen könnte. Nur Mut und fangt an!

```

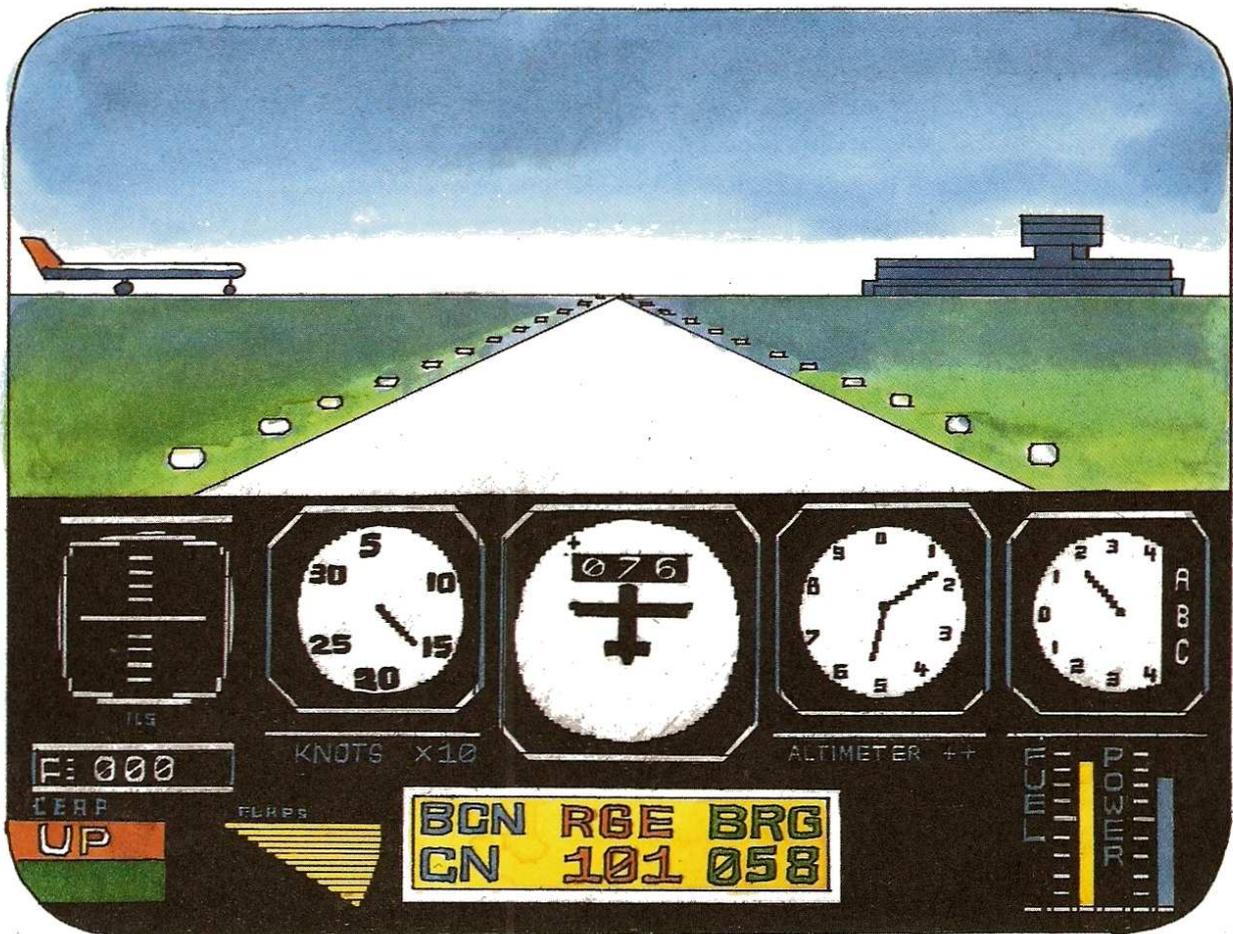
90 REM Ball
110 PRINT AT 9,19;"          "
120 PLOT 155,94 : PLOT 155,105 : PLOT 156,94 : PLOT 156,105
130 PLOT 153,95 : PLOT 153,104 : PLOT 154,95 : PLOT 154,104 : PLOT 156,95 :
    PLOT 156,104 : PLOT 157,95 : PLOT 157,104
140 PLOT 151,97 : PLOT 160,97 : PLOT 151,98 : PLOT 160,98 : PLOT 151,99 :
    PLOT 160,99 : PLOT 151,100 : PLOT 160,100 : PLOT 151,101 : PLOT 160,101 :
    PLOT 151,102 : PLOT 160,102
150 PLOT 150,99 : PLOT 161,99 : PLOT 150,100 : PLOT 161,100 : PLOT 150,101 :
    PLOT 161,101
160 STOP

```

Computerspiele

Wer von euch hat nicht schon einmal davon geträumt, selbst ein Flugzeug oder sogar ein Raumschiff zu steuern? Mit einem Computerprogramm könnt ihr euch diesen Traum fast wahr machen. Auf dem Bildschirm seht ihr das Cockpit des Flugzeuges mit dem Höhen- und Geschwindigkeitsmesser, der Anzeige für den Treibstoff, den Kontrolleuchten für das Fahrwerk, dem Kompaß und dem künstlichen Horizont. Außerdem könnt ihr noch durch das Fenster die Landschaft schräg vor euch sehen.

Wenn euer Computer jetzt noch einen JOYSTICK (joy – Spiel, stick – Stock; Steuerknüppel) hat, dann könnt ihr euch wie ein Pilot in der Flugzeugkanzel eines Sportflugzeuges fühlen. Die großen Düsenjets haben keinen Steuerknüppel, sondern so etwas wie ein halbes Lenkrad. Drückt ihr den Joystick nach vorn, senkt sich die Nase des Flugzeuges, die Erde rast auf euch zu, und der Motor heult auf. Zieht ihr den Knüppel zu euch heran, verschwindet der Horizont, und ihr schaut in die



Wolken oder fliegt sogar in der Rückenlage. Also aufgepaßt, daß ihr nicht abstürzt.

Der Lärm, den ihr beim Spielen hört, wird ebenfalls vom Computer erzeugt. Manche Computer benutzen dazu den Lautsprecher im Fernsehgerät, andere haben einen eigenen. Da ist die Tonqualität meist nicht besonders

gut. Der Computer kann aber nicht nur Lärm machen, sondern auch Musik. Der Befehl, der die Töne erzeugt, heißt BEEP (Pfiff). Bei einfachen Computern klingt die Musik sehr nach Blechbüchse. Es gibt aber auch Computer, die spezielle Teile zur Tonerzeugung haben, die man SYNTHESIZER nennt. Synthetisieren

heißt soviel wie künstlich zusammensetzen. Übrigens benutzt eure Lieblingspopgruppe für ihre Musik ebenfalls Synthesizer.

Wollt ihr den Roboter noch ein bißchen musizieren lassen? Dann fügt den Befehl 105 ein. Sofort gibt er bei jedem Auftauchen 3 mal $32 = 96$ Töne von sich.

105 BEEP 0.2,0 : BEEP 0.2,4 : BEEP 0.2,7
Die Länge des Tones wird durch die erste Zahl im BEEP-Befehl bestimmt. Die zweite Zahl gibt die Tonhöhe an. Welche Zahlen ihr einsetzt, hängt davon ab, wie euer Computer die Noten verschlüsselt.

Doch kehren wir zum Spiel mit dem Flugzeug zurück. Es gehört viel Geschick dazu, wenn das Flugzeug nach einem Flug über mehrere hundert Kilometer sicher landen soll. Da heißt es üben, genauso wie ein richtiger Pilot. Sie benutzen für ihr Training ähnliche Programme. Einen kleinen Unterschied gibt es allerdings dabei. Ihr sitzt auch bei einem Sturzflug ganz bequem in eurem Sessel. Der Pilot spürt dagegen jede Reaktion seines Computer-Flugzeuges und hängt demzufolge beim Sturzflug kopfüber in den Gurten seines Sessels.

Viele Computerspiele sind Geschicklichkeitsspiele. Es kommt bei ihnen darauf an, schnell zu reagieren, um möglichst viele Punkte zu sammeln oder zum nächsten Bild des Spieles zu gelangen. Das Prinzip dieser Spiele könnt ihr an einem Beispiel erkennen.

Ihr laßt den Computer an einer beliebigen Stelle und zu einer beliebigen Zeit für einen Moment auf dem Bildschirm ein Rechteck erzeugen. Dazu benutzt ihr wieder die Zufallszahl. Nun können verschiedene Aufgaben zu lösen sein. Wer benötigt die kürzeste Zeit, um nach dem Auftauchen des Rechtecks eine bestimmte Taste zu drücken? Oder wer schafft es rechtzeitig, mit einem anderen Zeichen dieses Rechteck zu treffen? Der Computer vergleicht die Zeiten und entscheidet, ob das Zeichen verschwindet und an einer anderen Stelle erneut auftaucht oder ob ihr erfolgreich wart.

Die Zeichen, mit denen ihr arbeitet, sind beliebig. Das Prinzip des Vergleichens ist aber bei diesen Geschicklichkeitsspielen immer wieder dasselbe. Auch bei den Spielen im Irrgarten. Euer Männlein befindet sich in einem Labyrinth und sucht einen Schatz. Auf dem Wege darf es keinem Monster begegnen. Geschieht das, wird euer Männlein auch zum Monster. Der Computer prüft nun ständig, ob die Positionen von Männlein und Monster übereinstimmen. Dabei bewegt er das Monster und entscheidet an den Ecken der Gänge mit einer Zufallszahl, in welche Richtung es weitergehen soll. Die Programme sind oft so geschrieben, daß sich das Monster etwas schneller bewegen kann als euer Männlein. Ihr seht, das Prinzip ist dasselbe wie bei dem eingangs geschilderten Beispiel.

Diese Spiele werden euch bestimmt eine gewisse Zeit an den Computer fesseln. Aber irgendwann wird euch diese Art der Spiele zu langweilig, obwohl es bei ihnen wunderschöne Bilder gibt. Wenn das der Fall ist, versucht es doch einmal mit ein paar Lernspielen. Ein ganz einfaches Programm könnt ihr sofort nachspielen. Es übt mit euch die Multiplikation der Zahlen von 1 bis 20. Wenn ihr wollt, dann erweitert das Programm gleich für die anderen Rechenarten.

Der Computer bestimmt zuerst die Zahlen, die miteinander multipliziert werden sollen. Nachdem ihr das Ergebnis eingetippt habt, prüft er, ob es richtig ist. War es falsch, druckt er das richtige Ergebnis und die entsprechende Multiplikationsreihe. Außerdem zählt er die Aufgaben und die richtigen Ergebnisse.

```
5 REM Rechne
10 LET L = 0 : LET N = 0
15 REM Zufallszahlen I und J
20 LET I = INT (RND 20) + 1 :
   LET J = INT (RND 20) + 1
25 LET N = N + 1
30 PRINT AT 10,10;I;" * ";J
40 INPUT "Ergebnis", K
```

```

45 REM Ergebniskontrolle
50 IF K<>I * J THEN 100
60 LET L = L + 1 :
   PRINT AT 0,6;N;"Aufgaben";
   L"Punkte"
70 PRINT AT 10,14;" = ";I * J : GOTO 20
100 PRINT AT 10,14;" = ";I * J :
110 PRINT AT 12,0;
   "Dein Ergebnis war leider falsch.
   Merke:"
115 CLS
120 FOR N = 1 TO I
130 PRINT AT M,0;M;" * ";J;" = ";M * J
140 NEXT M
150 INPUT "Alles klar? Ja = JA", A$
160 IF A$ <> "JA" THEN 150
170 CLS
180 GOTO 20

```

Wir haben in diesem Programm einen Befehl benutzt, den ihr noch nicht kennt. Mit Hilfe des Befehls 120 FOR (für) und TO (bis) sowie des Befehls 140 NEXT (der nächste) sparen wir viele Programmschritte. Alle Befehle, die von FOR und NEXT umschlossen werden, bei uns ist es nur der Befehl 130, bilden eine Schleife oder ein Nest. Nehmen wir an, die Zufallszahl I habe den Wert 15. Dann durchläuft der Computer die Schleife vom Befehl 120 bis 140 fünfzehnmal. Bei jedem Durchlauf wird der Wert von M um 1 erhöht. Mit dem Zählen wird in unserem Fall bei 1 begonnen. Nach dem 15. Durchlauf hat M den Wert 16. Aus dem Vergleich von M mit I stellt der Computer fest, daß die gewünschte Zahl von Durchläufen erfolgte, so daß er das Programm beim Befehl 150 fortsetzt.

Eine besondere Gruppe stellen die Denkspiele dar. Hier wählt der Computer nicht nur Zahlen oder Farben aus, die erraten werden müssen. Er spielt nach einem bestimmten Schema (ALGORITHMUS). Wenn ihr diesen Plan nicht kennt, habt ihr keine Chance, gegen den Computer zu gewinnen. Ihr könnt ihn also nur austricksen, wenn ihr seinen Spielalgorithmus

knackt. Bei den komplizierten Denkspielen ist das aber kaum möglich. Zu ihnen zählen die Schachprogramme. Obwohl es spezielle Schachcomputer gibt, könnt ihr auch mit eurem Mikrocomputer sehr gut Schach spielen. Diese Programme sind allerdings genauso wie die Spielprogramme nicht in Basic programmierbar. Die «Denkzeiten» des Computers würden sonst nicht nur wenige Minuten, sondern mehrere Stunden oder Tage dauern. Deshalb müssen die Programme in ihrer ganzen Länge in Maschinensprache übersetzt werden. Wird wie bei Basic immer nur ein Befehl übertragen und abgearbeitet, kostet das viel zu viel Zeit.

In den Schachprogrammen steckt monatelange Programmierarbeit. Benutzt der Programmierer dann sein eigenes Programm, wird er feststellen, daß der Computer besser als er selbst spielt. Er wird verlieren. Während der Computer 3 bis 4 Züge «vorausdenkt» und dabei alle möglichen Zugvarianten berechnet, begnügt sich ein Schachspieler mit einigen wenigen Varianten. Der Computer sucht unter den berechneten Zügen die beste Variante heraus. Außerdem hat er keine Konzentrationsschwierigkeiten. Er ermüdet nicht und läßt sich auch durch nichts ablenken. Das sieht bei uns schon ein bißchen anders aus.

Heute gibt es Schachprogramme mit verschiedenen Schwierigkeitsstufen. Sonst würden wir ja immer gegen den Computer verlieren. Je höher der Schwierigkeitsgrad des Programmes ist, um so länger braucht der Computer, um den Zug auszuwählen. Ihr werdet sicher nicht gleich mit der höchsten Stufe beginnen. Und wenn ihr trotzdem nicht gewinnt, dann grämt euch nicht. Nur jeder hundertste aller Schachspieler kann besser spielen als ein Computer mit einem modernen Schachprogramm.

LEXIKON



ADDIERSCHALTUNG

Schaltung zur Addition im Binärsystem. Ein 8-Bit-Mikroprozessor hat 8 Addierschaltungen auf dem Chip zur gleichzeitigen Durchführung der Rechenoperationen für jedes Bit. Diese Schaltungen bestehen wiederum aus anderen logischen Schaltungen (UND, ODER, NICHT).

ADRESSE

Kennzeichnet die Position eines Speicherplatzes im ROM und RAM oder einen Speicherbereich auf der Diskette. Die richtige Adressierung ist eine entscheidende Voraussetzung für die ordnungsgemäße Arbeitsweise eines Computers. Mit Hilfe von 16 Bit lassen sich $2^{16} = 65\,536$ Speicherplätze (64 KByte) adressieren.

ADRESSBUS

Verbindungsleitungen zwischen Mikroprozessor und RAM zur gleichzeitigen Übertragung der Speicherplatzadressen. Die Adressen werden mit Hilfe von Impulsen (ein = 1, aus = 0) verschlüsselt.

ALGORITHMUS

Vorschrift zum Lösen eines Problems. Nur wenn diese Vorschrift bekannt ist, kann ein Problem mit Hilfe eines Computers bearbeitet werden.

ALU

Arithmetik- und Logik-Einheit oder Rechenwerk (Arithmetic and Logic Unit) ist ein Teil des Mikroprozessors, in dem gerechnet (arithmetische Operationen) und verglichen (logische Operationen) wird.

AND-SCHALTUNG

siehe UND-SCHALTUNG

ARBEITSSPEICHER

In ihm werden Programme und Daten zur Abarbeitung für den Mikroprozessor gespeichert. Teile des Arbeitsspeichers sind der ROM (nur Lesespeicher) und der RAM (Schreib- und Lesespeicher). Er besteht aus einzelnen Zellen, die jeweils ein Byte speichern. Zu jedem Speicherplatz gehört eine Adresse. Die Speicherkapazität wird in Kilo-byte (KByte) angegeben. Um die Speicherkapazität des Computers zu erhöhen, gibt es außerdem

noch externe (äußere) Speicher, wie Kassetten oder Disketten.

ASCII-CODE

Amerikanischer Normcode für die Datenfernübertragung (American Standard Code for Information Interchange), dient zur binären Verschlüsselung der Eingabedaten und Entschlüsselung der Computerausgabe über den Drucker.

BASIC

(Beginners All-Purpose Symbolic Instruction Code.) Höhere Programmiersprache, die von allen Mikrorechnern verstanden wird. Sie ist leicht erlernbar und besteht nur aus einer relativ kleinen Zahl von Anweisungen. Da sehr viele Basic-Dialekte existieren, ist ein Austausch von Programmen verschiedener Computersysteme nicht ohne weiteres möglich.

BEFEHL

Anweisung, die vom Computer ausgeführt wird. Befehle sind Anweisungen zum Rechnen (10×17), zum Vergleichen ($17093 \stackrel{?}{=} 29876$), zum Übertragen von Daten ($A = B$), zum Einlesen oder zur Ausgabe von Daten und zu vielem anderem mehr. Die Gesamtheit der Befehle ist ein Programm. Ihre Abarbeitung erfolgt im Mikroprozessor.

BETRIEBSSYSTEM

Im ROM befindliche Programme, die für die Arbeit des Computers notwendig sind. Es erleichtert die Benutzung des Computers und organisiert das richtige Zusammenspiel aller zum Computer gehörenden Geräte.

BILDSCHIRM

Kann ein gewöhnliches Fernsehgerät oder ein speziell für den Computer angepaßtes Gerät zur Darstellung von Schrift oder Bildern sein. Er besteht aus vielen tausend Leuchtpunkten (Pixel), die vom Computer hell oder dunkel bzw. in der gewünschten Farbe eingestellt werden. Aus diesen Punkten setzt sich jedes Zeichen oder Bild zusammen.

BILDWIEDERHOLSPEICHER

Teil des RAM, in dem die Information für den Bildschirm gespeichert ist. Je mehr Farben und

Bildelemente für den Bildschirm benutzt werden, um so größer muß dieser Speicher sein.

BINÄRZAHL

Binär- oder Dualzahlen umfassen nur die Ziffern 0 und 1. Sie dienen zur Codierung von Zahlen, Buchstaben und Informationen.

BIT

Binäres Zeichen, welches zwischen zwei Zuständen (0 und 1, aus und ein, kurz und lang, hell und dunkel usw.) unterscheidet. Es dient als Maßeinheit der Informationsmenge.

BUS

Sammelschiene. Verbindungsleitungen zwischen Mikroprozessor und anderen Teilen des Computers zur Übertragung von Daten (Datenbus), Adressen (Adreßbus) oder Steuersignalen (Steuerbus).

BYTE

Stellt eine Folge von 8 Bit zur Codierung von Zeichen oder anderen Informationen dar. Es dient als Maßeinheit von Informationsmengen.

1 Byte = 8 Bit, 1 KByte = 2^{10} Byte = 1024 Byte

CAD

Rechnergestütztes Konstruieren oder Entwerfen (Computer Aided Design) von Werkstücken, Modeartikeln, Maschinen, Häusern usw.

CAM

Rechnergestützte Fertigung oder Produktion (Computer Aided Manufacturing) aller nur denkbaren Produkte.

CHIP

Halbleiterplättchen aus Silizium, auf dem sich Tausende von integrierten Schaltkreisen befinden.

CODE

Zuordnung von Zeichen, Ziffern und Buchstaben zu Kombinationen von Nullen und Einsen, z. B. der Buchstabe E: 01100101.

COMPILER

Übersetzt das gesamte Programm, das in einer

höheren Programmiersprache geschrieben wurde, in die Maschinensprache des Computers. Nach der Übersetzung wird das Programm abgearbeitet, wofür wesentlich weniger Zeit benötigt wird als für ein Programm, welches mit einem Interpreter bearbeitet wird.

CPU

Zentraleinheit (Central Processing Unit)

siehe Mikroprozessor

CURSOR

Positionsanzeiger, der die Stelle auf dem Bildschirm angibt, an der das nächste Zeichen bei einer Eingabe erscheint. Die Verschiebung in beliebige Richtungen erfolgt mit Hilfe von 4 Tasten.

DATEN

Werte und Angaben, die einen Sachverhalt beschreiben.

DISKETTE

Kunststoffplatte (floppy disk – biegsame Scheibe), auf der sich eine Magnetschicht befindet. Auf ihr werden ähnlich wie auf einer Magnetbandkassette Programme und Daten unter einem Namen aufgezeichnet und gesichert. Die Namen dienen zur schnellen Auffindung der Programme oder Daten. Die Disketten haben überwiegend einen Durchmesser von $5\frac{1}{4}$ Zoll (ca. 13 cm) und werden in speziellen Disketten-Laufwerken vom Computer beschrieben und gelesen. Ihre Speicherkapazität beträgt zwischen 150 und 800 KByte. Es gibt außerdem Disketten von $3\frac{1}{2}$ und 8 Zoll Durchmesser.

DRUCKER

Ausgabegerät des Computers. Wenn er wie eine Schreibmaschine funktioniert, werden nur Buchstaben, Zahlen und Sonderzeichen gedruckt. Naddrucker setzen sämtliche Zeichen mit Hilfe eines Rasters aus einzelnen Punkten zusammen, deren Feinheit von der Größe der Nadeln bestimmt wird. Damit lassen sich auch Zeichnungen anfertigen.

DUALZAHL

siehe Binärzahl

FEHLERMELDUNG

Falsche Befehle, unlogische Schritte im Programm oder Fehlbedienungen werden vom Computer erkannt und auf dem Bildschirm angezeigt.

FESTSPEICHER

siehe ROM

FORTRAN

Wörtlich Formelübersetzer (Formula Translator). Älteste höhere Programmiersprache für wissenschaftlich-technische Aufgaben. Die Sprache wurde inzwischen mehrfach weiterentwickelt und trägt heute die Bezeichnung FORTRAN IV.

FREAK

Begeisterter Mensch, der die Funktionsweise eines Gerätes sehr gut kennt und dadurch intensiv nutzen kann (Computerfreak).

HARDWARE

Wörtlich harte Ware. Sammelbezeichnung für alles, was am Computer aus Material ist (Bauelemente, Bausteine, Geräte usw.). Die Programme zur Verwendung der Hardware nennt man im Gegensatz dazu Software (weiche Ware).

IMPULS

Mit Hilfe elektrischer Impulse werden Zeichen verschlüsselt, Informationen übertragen oder Rechnungen im Computer ausgeführt.

INFORMATION

Stellt den Inhalt einer übermittelten Nachricht dar. Wie groß die Informationsmenge ist, wird in Bit oder Byte angegeben.

INTERFACE

Wörtlich Zwischengesicht. Ein als Schnittstelle bezeichneter Übergang zwischen unterschiedlichen Teilen eines Systems, z. B. zwischen Computer und Drucker oder Computer und Kassettenrecorder.

INTERPRETER

Wörtlich Dolmetscher. Überträgt Befehl um Befehl eines Programms, das z. B. in Basic geschrieben wurde, vor dessen Ausführung in die Maschinensprache des Computers. Das hat den Vorteil,

daß man die Abarbeitung des Programms unterbrechen kann, danach Befehle oder Daten verändert und dann das Programm an der Stelle, wo es unterbrochen wurde, fortsetzen kann. Dabei gehen keinerlei Zwischenergebnisse verloren. Die fortlaufende Übertragung der Befehle in die Maschinensprache kostet allerdings bei der Abarbeitung des Programms Zeit. Bei einem komplizierten Programm wird diese Zeit durch Benutzung eines Compilers eingespart.

JOYSTICK

Steuerknüppel zur Bewegung von Figuren in beliebige Richtungen auf dem Bildschirm. Erleichtert Computerspiele.

KASSETTE

Auf Magnetbandkassetten können vom Computer über einen gewöhnlichen Kassettenrecorder Programme und Daten aufgezeichnet und von dort wieder gelesen werden. Diese werden fortlaufend aufgezeichnet, so daß sowohl das Auffinden als auch das Lesen wesentlich langsamer erfolgt als bei Disketten.

KOMMANDO

Steuert den Computer direkt von der Tastatur aus, so daß seine Benutzung sehr erleichtert wird. Wichtige Basic-Kommandos sind RUN zum Starten eines Programms, SAVE zum Sichern von Programmen oder Daten auf Diskette oder Kassette, LOAD zum Laden derselben von dort in den Computer, BREAK zum Unterbrechen eines laufenden Programms und andere.

LICHTSTIFT

Eingabegerät, z. B. zur Erzeugung von Bildern auf dem Bildschirm oder zur Übertragung von Zeichnungen in den Computer.

MASCHINENSPRACHE

Programmiersprache, deren Befehle für den Mikroprozessor direkt verständlich sind. Sie ist nicht leicht anwendbar. Deshalb werden Programme in höheren Programmiersprachen geschrieben und vom Computer mit einem Compiler in diese übersetzt oder durch einen Interpreter Befehl um Befehl in die Maschinensprache übertragen.

MAUS

Rollkugelinterface zur Dateneingabe (z. B. bei Computerspielen oder für Zeichnungen)

MIKROPROZESSOR oder CPU

Organisiert die Arbeit des Computers. Er sorgt für die Abarbeitung der Befehle, rechnet, vergleicht und kontrolliert alle Tätigkeiten. Es gibt Prozessoren, die 8, 16 oder 32 Bit bei einer Operation gleichzeitig verarbeiten können. Er besteht aus dem Rechen- und dem Steuerwerk.

MODEM

Gerät zur Datenübertragung von einem Computer per Telefon zu einem anderen Computer, der viele tausend Kilometer entfernt sein kann.

MODUL

Teil des Ganzen. Mit Hilfe von Modulen kann z. B. die Speicherkapazität des Computers erweitert werden. Es gibt auch Programm-Module, die z. B. Computerspiele enthalten.

NICHT-SCHALTUNG oder NOT-SCHALTUNG

Grundschialtung der Elektronik. Sie hat einen Eingang und einen Ausgang. Sie wandelt die Impulse um, d. h. aus 0 wird 1 und aus 1 eine 0. Sie ist Bestandteil der Addierschialtung.

ODER-SCHALTUNG, auch OR-SCHALTUNG

Grundschialtung der Elektronik mit zwei Eingängen und einem Ausgang. Am Ausgang ergibt sich nur dann eine 0, wenn an beiden Eingängen eine 0 anliegt. Bestandteil der Addierschialtung der CPU.

PC oder PERSONALCOMPUTER

Computer, die im allgemeinen für einen speziellen Arbeitsplatz vorgesehen sind.

PLATINE

Leiterplatte, auf der sich sämtliche Bausteine des Mikrorechners befinden. Sie wird über Steckkontakte mit der Stromversorgungseinheit des Computers und den anderen außerhalb des Gehäuses befindlichen Geräten verbunden.

PIN

Kontaktstift

PIO

Tor zur gleichzeitigen Ein- und Ausgabe (Parallel Input Output) von Impulsen in den bzw. aus dem Computer. Bei einem 8-Bit-Prozessor liegen stets 8 Bit gleichzeitig (parallel) am Tor an. Im Gegensatz dazu: SIO.

PIXEL

Bildelement (Picture element). Bezeichnung für die einzelnen Leuchtpunkte auf dem Bildschirm.

PLOTTER

Ausgabegerät, zur Anfertigung von Zeichnungen. Ein Tintenstift mit unterschiedlichen Farben wird dazu vom Computer in alle gewünschten Richtungen über das Papier geführt.

PROGRAMM

Sinnvolle numerierte Anordnung von Befehlen. Es teilt dem Computer mit, in welcher Reihenfolge die Befehle abgearbeitet werden sollen. Es wird in einer Programmiersprache geschrieben. Zur Abarbeitung befindet es sich im RAM.

PROGRAMMIERSPRACHE

Besteht aus einer Sammlung von Befehlen und Regeln, wie diese Befehle zu schreiben sind. Die Befehle der höheren Programmiersprachen BASIC, FORTRAN, COBOL, PL1 und anderer müssen vor der Abarbeitung in die Maschinensprache des Computers übertragen werden. Das geschieht mit speziellen Programmen (Compiler oder Interpreter).

RAM

Speicher mit beliebigem Zugriff (Random Access Memory). Lese- und Schreibspeicher, bei dem jeder Speicherplatz adressierbar ist. In ihm werden Programme und Daten während des Betriebs des Computers gespeichert. Beim Abschalten des Computers werden alle Speicherplätze gelöscht.

RASTER

Schema zur Erzeugung von Zeichen oder Bildelementen.

RECHENWERK

siehe ALU

ROM

Nur-Lese-Speicher (Read Only Memory), in dem vom Computerhersteller für den Betrieb des Computers notwendige Befehle gespeichert sind. Eine Korrektur des Speicherinhalts ist nicht möglich. Der ROM ist besonders geschützt.

SIO

Schnittstelle zur jeweils nacheinander erfolgenden (Serial Input Output) Ein- bzw. Ausgabe von Impulsen in den oder aus dem Computer (z. B. zum Kassettenrecorder). Im Gegensatz dazu PIO.

SOFTWARE

Wörtlich weiche Ware. Sammelbezeichnung für alles am Computer, was nicht als Gerät betrachtet werden kann. Dazu gehören insbesondere die Programme.

SONDERZEICHEN

Auf der Tastatur befindliche Zeichen, z. B. + - / ! & () :

SPEICHERKAPAZITÄT

Gibt an, wieviel Zeichen oder Byte die einzelnen Speicher aufnehmen können. Arbeitsspeicher haben eine Kapazität von 64 KByte bis zu mehreren MByte (M = Mega: Million). Disketten haben eine Speicherkapazität zwischen 150 KByte und 800 KByte.

STEUERBUS

Verbindungsleitungen zwischen Mikroprozessor und den anderen Bausteinen und Geräten zur Übertragung von Steuersignalen.

STEUERWERK

Bestandteil des Mikroprozessors. Es steuert alle Funktionen des Computers und überwacht die externen (äußeren) Geräte.

SYNTHESIZER

Gerät zur Erzeugung von Tönen, Lauten oder Sprache.

TAKTFREQUENZ

Häufigkeit der Impulse. Sie ist für die Schnelligkeit der Arbeit des Computers mitverantwortlich. Bei

einer Taktfrequenz von 2 MHz (2 Megahertz = 2 Millionen Taktimpulse pro Sekunde) führt der Mikroprozessor ungefähr 250 000 Operationen aus.

TAKTGEBER

Innere Uhr des Computers. Er erzeugt Impulse im Abstand von weniger als einer Mikrosekunde (1 Millionstel Sekunde) oder mehr als 1 Million Impulse pro Sekunde. Mit Hilfe dieser Impulse werden sämtliche Operationen des Computers gesteuert.

TASTATUR

Besteht aus Tasten für Zahlen, Buchstaben, Sonderzeichen und Funktionen. Über die Tastatur werden Kommandos, Befehle und Daten eingegeben. Sie ist das wichtigste Gerät für die Verständigung mit dem Computer.

UND-SCHALTUNG oder AND-SCHALTUNG

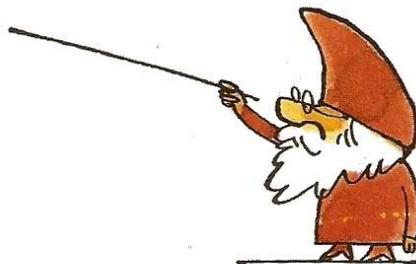
Grundschialtung der Elektronik mit zwei Eingängen und einem Ausgang. Am Ausgang ergibt sich nur dann eine 1, wenn an beiden Eingängen eine 1 anliegt. Sonst ist der Ausgang 0. Bestandteil der Addierschialtung.

ZENTRALEINHEIT (CPU)

siehe Mikroprozessor

ZUFALLSZAHL

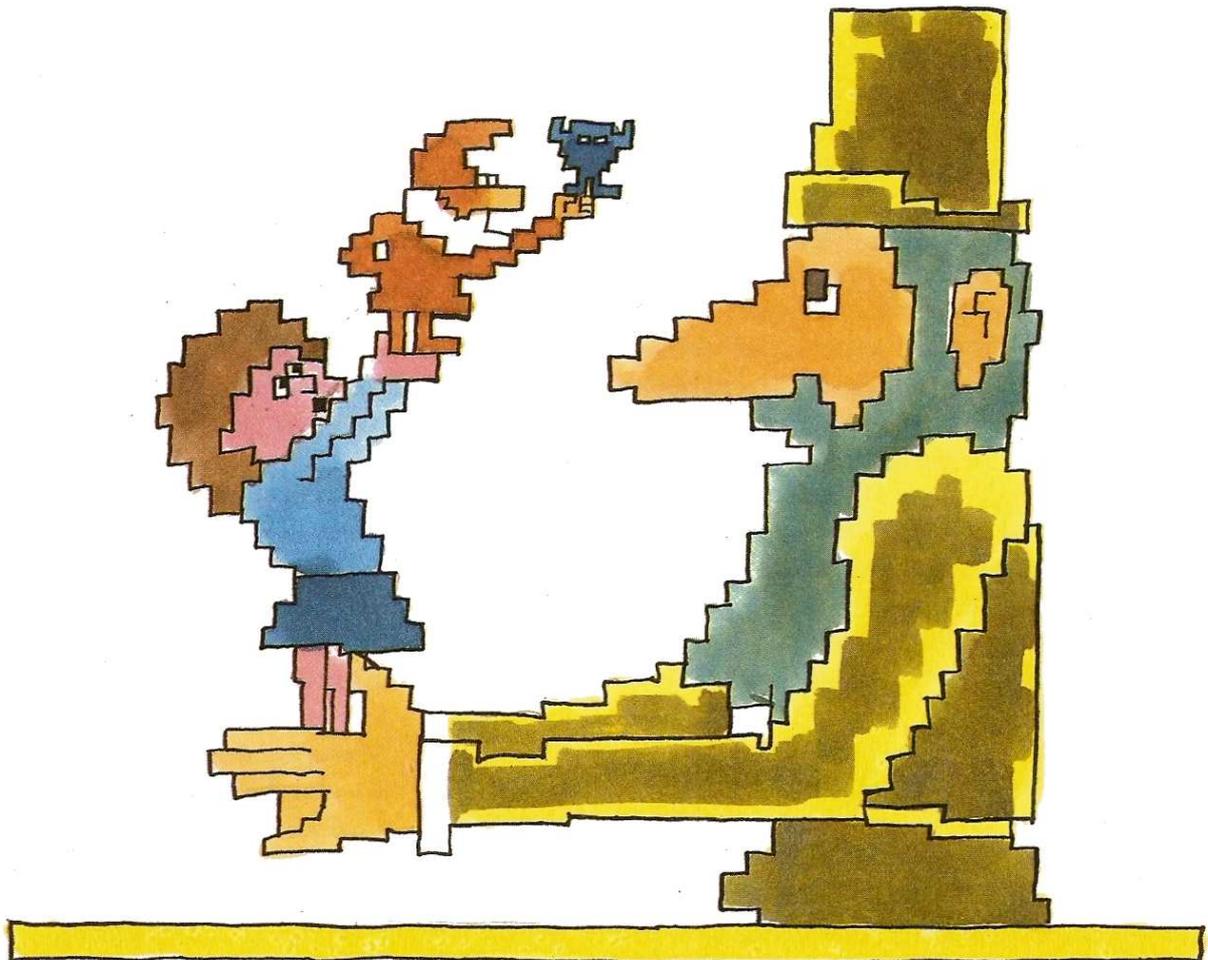
Zahl, die durch ein spezielles Programm berechnet wird und nicht vorausgesagt werden kann.





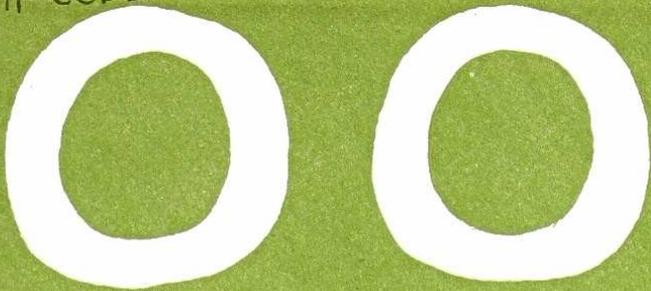
ISBN 3-7302-0529-3
© 1989 Verlag Junge Welt Berlin/DDR
1. Auflage
Lizenznummer: 308/21/89
LSV 9990

Printed in the German Democratic Republic
Gesamtherstellung: Grafischer Großbetrieb Sachsen-Druck Plauen
Bestellnummer: 683 738 3
Lektor: Irene Kahlau
Gestalter: Sylvia Brendel
Für Leser von 9 Jahren an
01200

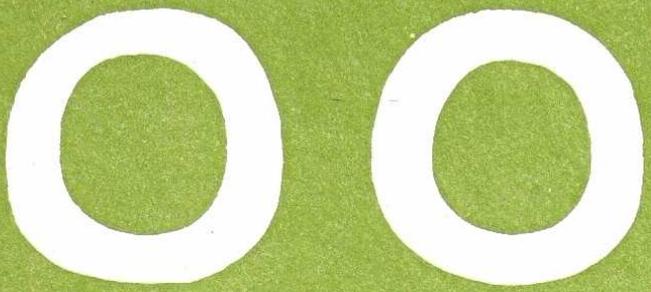


ASCII-CODE

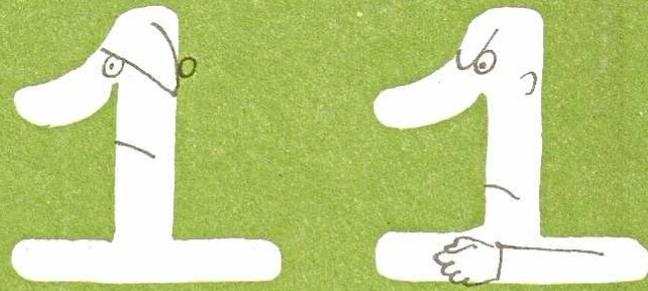
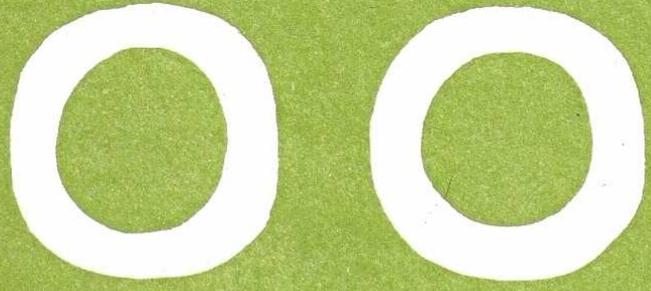
0



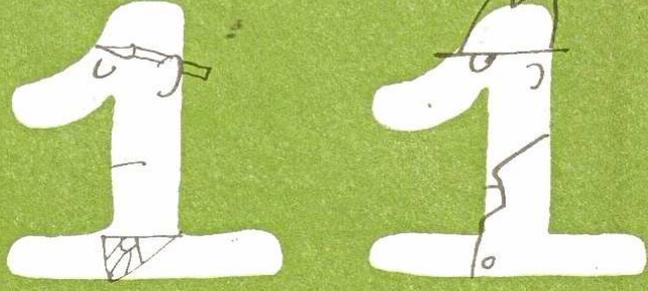
1



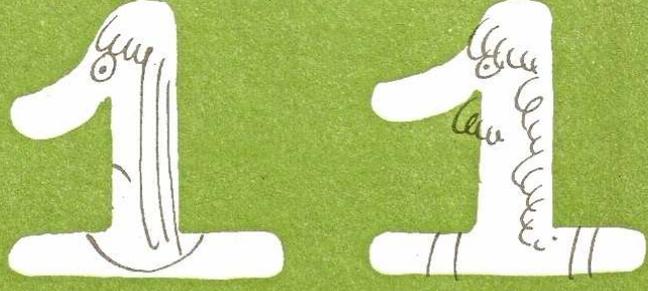
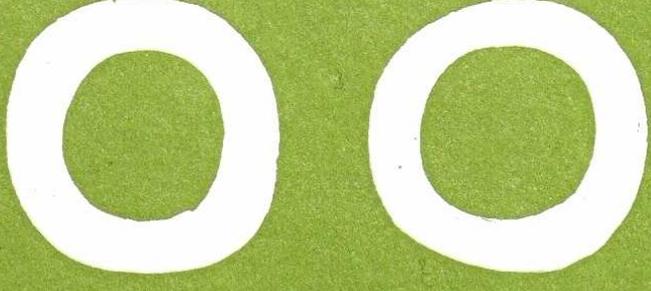
2



3

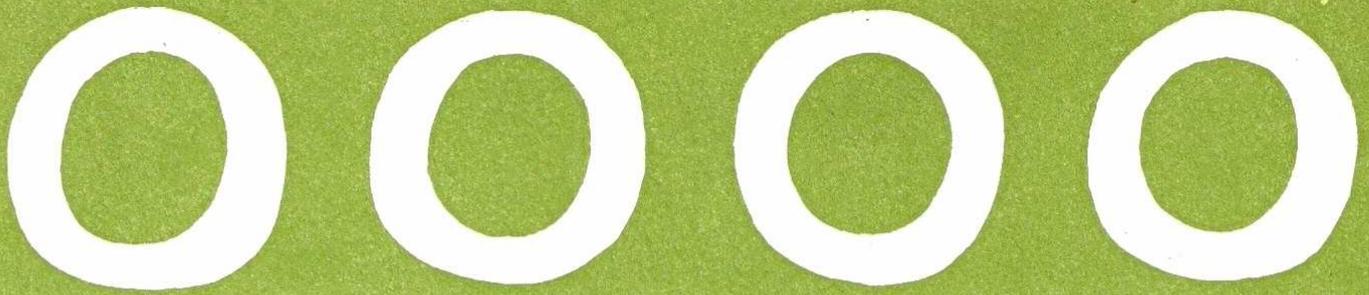


4

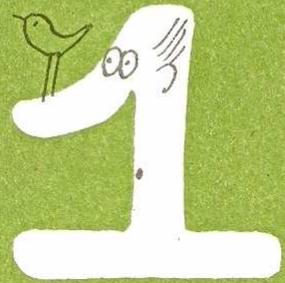
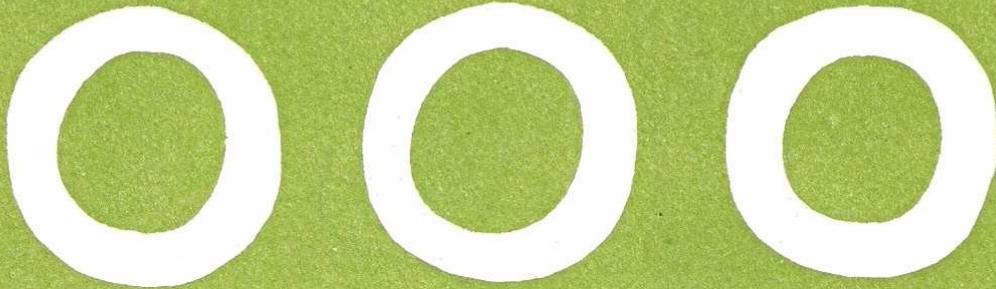


5

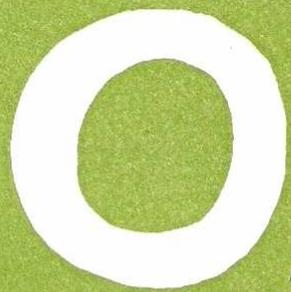
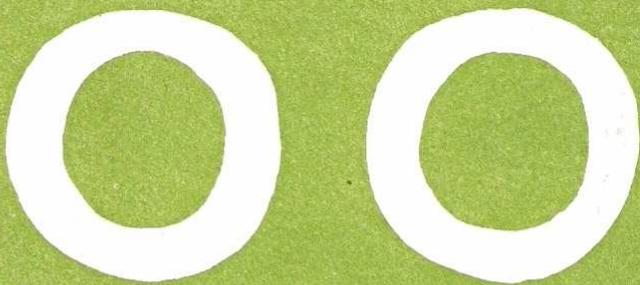




0



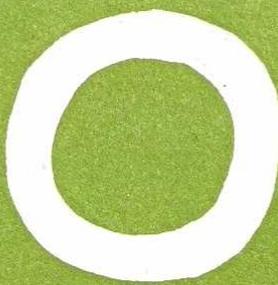
1



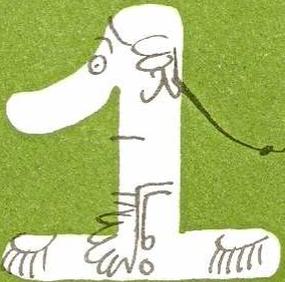
2



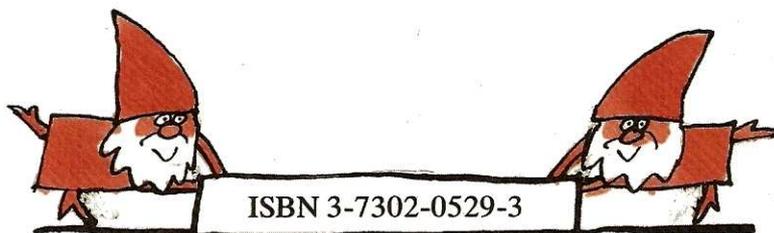
3



4



5



ISBN 3-7302-0529-3