

PC1260

**Ger mann
Jörg
Waldvogel**

**Das große
Pocket
Computer
Buch**

PC 1261

SHARP

PC1401

PC1402

PC 1421

PC1350

Ein DATA BECKER Buch

Do not sale!

ISBN 3-89011-173-4

**Copyright © 1986 DATA BECKER GmbH
Merowingerstraße 30
4000 Düsseldorf**

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

Do not sale !

Wichtiger Hinweis:

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

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

Do not sale !

Do not sale !

Vorwort

Liebe Leserin, lieber Leser!

Die Leistungsfähigkeit der Sharp Taschencomputer ist heute unbestritten. Es sind handliche Computer, die mit einem BASIC ausgestattet sind, von dem sich einige der größeren Kollegen eine Scheibe abschneiden könnten.

Leider ist die Freude der Besitzer nicht ungetrübt, denn trotz sehr großer Verbreitung, sind relativ wenig Zusatzinformationen und wenig Software für diese Rechner erhältlich. Diese Tatsache läßt sich durch die kleinen Unterschiede in der Konzeption der einzelnen Geräte der Sharp PC Reihe erklären.

Zwar sind die Computer der einzelnen Serien relativ gleich gestaltet, aber in den seltensten Fällen kann man Programme, die für den PC 1350 geschrieben worden sind, ohne Änderungen auf einem PC 1246 nutzen. Dieser Umstand ergibt sich zum einen aus der größeren Speicherkapazität des PC 1350, und zum anderen aus dem größeren BASIC-Befehlssatz des PC 1350. Diese Problematik läßt sich im Grunde auf alle Sharp PC übertragen. Eine Lösung dieser Problematik bietet ein Standard, der die Teilkompatibilität der einzelnen Rechner untereinander ausschließt, und somit Programme ermöglicht, die auf allen Sharp Pocket Computern nutzbar sind.

Dieser Standard, der SPCS (Sharp Pocket Computer Standard) wird in diesem Buch eingeführt und erläutert. Programme, die unter Beachtung der Richtlinien dieses Standards geschrieben worden sind, können auf allen Sharp PC Rechnern ohne Änderung übernommen werden.

Dabei ist eine kleine Einschränkung zu machen. Sharp PC, die unter den Möglichkeiten des PC 1246 liegen, sind nicht im Standard enthalten. Für die, die nicht über die technischen Daten des PC 1246 informiert sind, sei kurz angemerkt, daß der PC 1246 über eine Speicherkapazität von 1278 Bytes und ein

Do not sale !

einzeiliges 16 Zeichen umfassendes Display verfügt. Alle Sharp PC, die mindestens diese Werte aufweisen, sind SPCS-tauglich. Um diesen Standard zu nutzen muß man lediglich die Richtlinien, die der SPCS vorgibt, einhalten.

Der SPCS enthält einen Hardware- und einen Softwareteil. Im Hardwareteil werden die Voraussetzungen erläutert, die es ermöglichen, die Unterschiede in der Hardware der einzelnen Rechner zu kompensieren.

Der Softwareteil gibt eine Übersicht über die im SPCS enthaltenen BASIC-Befehle. Die Befehle sind nach Funktionsbereichen gegliedert und kurz erläutert. Die Bemühungen, den theoretischen Teil so prägnant wie möglich zu halten, kommt sicherlich all denjenigen zugute, deren Interesse der Anwendung an sich gilt.

Im Blickpunkt dieses Buches liegt der Programmteil, in dem Anwendungen verschiedenster Art im SPCS gelistet und erläutert sind. Die Anwendungen reichen vom Lösen linearer Gleichungssysteme über Abschreibungen bis zum einarmigen Banditen.

Zu jedem Programm gehört ein einleitender und erklärender Text, eine Variablenliste, das Listing und ein Vorschlag, der Ihnen Anregungen zur Verbesserung und Erweiterung der bestehenden Programme gibt.

Dieser Vorschlag ist neben dem Standard ein weiteres Kriterium, das dieses Buch von einer Programmsammlung im bekannten Sinne abhebt. Es soll Ihnen nicht nur fertige Software bieten, sondern Ihre Kenntnisse und Fertigkeiten der BASIC-Programmierung festigen und erweitern.

Viel Spaß dabei wünscht Ihnen

Holger Brechmann

Do not sale !

Inhalt

1.	PC = Computer?	11
2.	BASIC-Programme	17
2.1	Schule und Studium	19
2.1.1	Brüche	21
2.1.2	Wie alt ist Tante Frieda?	22
2.1.3	Körper	26
2.1.4	ggT und kgV	28
2.1.5	Vokabelpauken	30
2.1.6	Pythagoras	34
2.1.7	Mathe-Trainer	35
2.1.8	Ohmsches Gesetz	38
2.1.9	Pfuschat	40
2.2	Programme für den Beruf	43
2.2.1	Abschreibungen	45
2.2.2	Calc	47
2.2.3	Yards und Inches	50
2.2.4	Yards - Dezimal	53
2.2.5	Yards und Meter	55
2.2.6	Optimal	58
2.2.6	mph - km/h - kn	60
2.2.7	Mittelwert	62
2.2.8	Währungen	64
2.2.9	Währungen 2	67
2.3	Programme für den Haushalt	69
2.3.1	Übergewicht??	71
2.3.2	Sprit	74
2.3.3	Telefon-Takt	75
2.3.4	Restwertberechnung	78
2.3.5	Währungen	79
2.3.6	Lottozahlen	82
2.3.7	Lottozahlen 2	83

Do not sale !

2.3.8	Zinseszins	84
2.3.9	Darlehen	86
2.4	Programme zur Animation	87
2.4.1	Von Rechts nach Links.....	89
2.4.2	Super Kegel-Spiel.....	90
2.4.3	Roulette Spezial	93
2.4.4	Die Zahl PI	94
2.4.5	Das Orakel	96
2.4.6	Tage.....	98
2.4.7	Temperaturen.....	101
2.5	Programme für das Hobby	103
2.5.1	Monitor	105
2.5.2	Weltweit.....	107
2.5.3	RND-RND.....	109
2.5.4	RND Test.....	111
2.5.5	Space Shuttle.....	112
2.5.6	Systeme.....	114
2.5.7	Bubble	115
2.5.8	Primzahlen	117
2.6	Programme zum Spielen.....	119
2.6.1	Henker.....	121
2.6.2	Jackpot	124
2.6.3	Zahlenschloß	126
2.6.4	Rekneh	129
2.6.5	Das Kim-Spiel	131
2.6.6	Turf	134
2.6.7	High Noon	137
2.6.8	Torten.....	139
2.6.9	U-Boot	141
2.6.10	Frosch	144
3.	Der Sharp-PC-Standard.....	147
3.1	Das Standard BASIC	149
3.1.1	Numerische Funktionen	149
3.1.2	Stringfunktionen.....	153

Do not sale !

3.1.3	Logische Operatoren.....	157
3.1.4	Programm-Anweisungen.....	157
3.1.5	Ein- und Ausgabeanweisungen	164
3.1.6	Programmspeicherung	169
3.1.7	Druckerausgabe.....	170
Anhang A	Alphabetische Programmliste	173
Anhang B	ASCII-Tabelle.....	175
Anhang C	Alphabetisches Verzeichnis der BASIC-Befehle.....	177

Do not sale !

1. PC = Computer?

Do not sale !

Do not sale !

Ist mein Sharp Pocket Computer denn nun ein richtiger Computer oder handelt es sich bei diesem Gerät nur um einen besseren Taschenrechner? Diese Frage beschäftigt jeden, der einen Sharp PC besitzt oder die Anschaffung eines solchen Gerätes plant. Um auf diese Frage eine klare Antwort geben zu können, müssen wir zunächst der Problematik an sich auf den Grund gehen. Die Firma Sharp nennt ihre Geräte Pocket Computer, also Taschencomputer. Die Namensgebung ist für jeden, der einen Sharp PC schon einmal in der Hand gehabt hat, sehr leicht nachvollziehbar. Ein Sharp PC paßt wirklich in die Hosenbeziehungweise in die Jackentasche. Allein der Begriff Computer im Namen könnte eine Angriffsfläche für Leute geben, die diese Geräte als "nettes Spielzeug" bezeichnen. In einem allseits anerkannten Microcomputer-Lexikon ist der Begriff Computer wie folgt definiert:

Computer = Allzweck-Rechensystem bestehend aus CPU, Speicher, E/A-Anschlüssen, Stromversorgung und Schrank.

Wenn wir einen Sharp PC auf die Definition hin überprüfen, fällt sehr schnell auf, daß der Pocket Computer fast alle Merkmale der Definition erfüllt. Das einzige Kriterium, das der Sharp PC nicht zu erfüllen scheint, ist der Schrank. Die Frage ist natürlich: Wozu braucht ein so kompaktes Computersystem einen Schrank?

Als Schlußfolgerung aus dieser Definition kann man schon einmal schließen, daß ein Sharp PC ein richtiger Computer ist. Beim Blättern in dem Lexikon ist mir dann mehr aus Zufall eine weitere Definition in die Hände gefallen. Die Definition bezieht sich auf die Bedingungen, die ein Personal-Computer erfüllen muß:

Personal-Computer = Billiger und leicht zu transportierender Computer, dessen Software auf einfache Ein-Benutzer-Anwendungen ausgerichtet ist.

Wenn eine Maschine preiswert (das Wort "billig" hat einen etwas negativen Beigeschmack) und leicht zu transportieren ist, dann doch wohl ein PC. Auch die Software eines PC ist auf einfache

Ein-Benutzer-Anwendungen ausgerichtet. Offensichtlich ist dieses Gerät nicht nur ein Computer im weitesten Sinne, sondern der Inbegriff eines Personal-Computers.

Nach diesen etwas überraschenden Ergebnissen sollten wir nicht vergessen, daß ein Sharp PC Einschränkungen unterworfen ist, die nicht von der Hand zu weisen sind. Das Display der kleineren Pocket Computer ist für viele Anwendungen nicht besonders geeignet. Wer versucht, Texte mit einem PC 1246 zu bearbeiten, der wird dieser Tatsache sehr schnell ins Gesicht sehen müssen. Auch für graphische Darstellungen ist das Display der einzeiligen PC nicht eingerichtet. Anders verhält es sich beim PC 1350, PC 2500 und dem PC 1600. Auf diesen Rechnern lassen sich auch Graphiken in beschränktem Maße, aber durchaus sinnvoll realisieren. Zudem sollte man nicht vergessen, daß es auch für die kleinen PC sehr portable Drucker und Plotter gibt, die graphische Darstellungen auch für diese Geräte möglich machen.

Das Argument des zu kleinen Arbeitsspeichers wird, obwohl es beim PC 1246 sicher seine Berechtigung hat, durch die technischen Daten des PC 1600 völlig entkräftet. Der PC 1600 bietet im voll ausgebauten Zustand 80 kBytes RAM! Ein weiteres wichtiges Kriterium für einen modernen Computer ist seine Benutzerfreundlichkeit in Bezug auf seine Programmiersprachen. Dazu ist in Hinblick auf Sharp Pocket Computer folgendes zu sagen. Alle Sharp PC sind von Haus aus mit einem sehr guten BASIC ausgestattet. Der Befehlsvorrat ist im Vergleich zu anderen Geräten zwar etwas kleiner, was aber mit fehlenden Graphikbefehlen zu erklären ist. Weiterhin verfügen alle Rechner dieser Sorte über einen PEEK- und einen POKE-Befehl. Wie jeder kundige Leser weiß, bedeutet das Vorhandensein dieser Befehle die Möglichkeit der Programmierung in Maschinensprache. Für den bisher verbreitetsten PC, den PC 1500, gibt es seit Mitte 1985 sogar einen FORTH-Compiler/Interpreter.

Zieht man nun alle in diesem Kapitel genannten Tatsachen in Betracht, so ergibt sich tatsächlich eine Antwort auf die zu Beginn gestellte Frage. Die Pocket Computer der Sharp PC Reihe sind Computer, die über alle Merkmale eines vollwertigen Computers verfügen. Allerdings gibt es Bereiche, in denen Sie

Do not sale !

Home- und Personal Computern deutlich unterlegen sind. Diese Unterlegenheit ergibt sich aus der speziellen Konzeption dieser Geräte. Es sind kleine, aber vollständige Computersysteme, die wirklich als portabel bezeichnet werden können. Sie sind eigenständige Systeme, die aber auch als Ergänzung zu Tischrechenanlagen dienen können. In diesem Sinne können Sie mit ruhigem Gewissen sagen, daß Sie im Besitz eines richtigen Computers sind, der nebenbei auch noch als wissenschaftlicher Taschenrechner benutzt werden kann.

Achtung: Aus drucktechnischen Gründen wird in diesem Buch der "Hochpfeil" als "^" dargestellt.

Do not sale !

2. BASIC-Programme

Do not sale !

Do not sale !

2.1 Programme für Schule und Studium

Die Programme in diesem Kapitel sind, wie aus der Überschrift schon ersichtlich, für die Bereiche Schule und Studium gedacht. Natürlich kann man im Rahmen einer Programmsammlung nur einige Aspekte der nahezu endlosen Anwendungsmöglichkeiten in diesen Bereichen ansprechen, zumal in dieser Programmsammlung der Bereich Schule und Studium nur einer von 6 Themenbereichen ist. Dementsprechend sind auch die Themen der Programme weit gestreut. Die Themen der 9 Programme in diesem Kapitel werden nun im einzelnen kurz erläutert.

1. *Brüche*

Dieses Programm kürzt Brüche so weit wie möglich und gibt zusätzlich den Bruch als Dezimalzahl aus. Gedacht ist das Programm für Schüler der unteren Klassen, die sich unter anderem mit dieser Problematik herumschlagen müssen.

2. *Frieda*

Der Name des Programmes ist etwas irreführend, denn Frieda ist ein Programm, mit dessen Hilfe man Gleichungssysteme mit drei Unbekannten lösen kann. Besonders Schüler der Mittel- und Oberstufe werden die Leistungen dieses Programms sehr zu schätzen wissen.

3. *Körper*

Volumen und Oberfläche von 5 verschiedenen Körpern können mit diesem menügesteuerten Programm berechnet werden. Auch dieses Programm ist im Unterricht und bei Hausaufgaben eine wertvolle Hilfe.

4. ggT und kgV

Vor allem Schüler des fünften und sechsten Schuljahres werden über dieses kleine aber leistungsstarke Programm hocheifrig sein. Der Rechner gibt sofort den größten gemeinsamen Teiler und das kleinste gemeinsame Vielfache zweier Zahlen an.

5. Vokabeln

Vokabellernen ist mit einem Computer einfacher, angenehmer und oft auch leichter als mit herkömmlichen Mitteln. Die schwierigen Vokabeln vor der Englischarbeit, oder auch die letzten Formeln vor der Prüfung lassen sich mit diesem Programm schnell noch einpauken.

6. Pythagoras

Der Satz des Pythagoras ist eine häufigt gebrauchte Formel, die nun in einem bequemen Programm vorliegt. Nur die Werte eingeben und schon ist das Ergebnis da.

7. Mathetrainer

Damit man das Kopfrechnen durch die vielen bequemen Programme nicht verlernt, hilft der Mathetrainer jederzeit, die grauen Zellen in Schwung zu halten. Mathetrainer bietet das Training der vier Grundrechenarten in zwei Schwierigkeitsstufen.

8. Ohmsches Gesetz

Eines der wichtigsten Gesetze der Elektrizitätslehre als Programm. Sobald man 2 Werte hat, hat man auch das Ergebnis. Dabei spielt es keine Rolle um welche beiden der drei Werte es sich handelt.

9. Pfuschomat

Das besondere Bonbon zum Schluß dieses Kapitels. Pfuschomat ist ein universelles Pfuschprogramm, mit einer eingebauten Lehrersicherung. 30 kurze Merksätze werden von Pfuschomat verwaltet und zu rechten Zeit auf das Display gebracht.

2.1.1 Brüche

Das Kürzen von Brüchen ist für viele eine der unangenehmen Aufgaben in der Schulmathematik. Diese Einstellung läßt sich allerdings sehr schnell ändern.

Mit dem nachfolgenden Programm ist das Kürzen von Brüchen eine Ihrer leichtesten Übungen. Das Programm kürzt jeden Bruch und gibt den gekürzten Bruch und den Kürzwert aus. Danach wird der Dezimalbruch berechnet und ausgegeben.

Variablenliste:

D:	Dezimalzahl
N:	Nenner
R:	Rest
X:	Zwischenspeicher
Y:	Zwischenspeicher
Z:	Zähler

```
1 CLEAR
2 WAIT 150
3 PRINT"KUERZEN VON"
4 PRINT"BRUECHEN"
5 PRINT" IM SPCS"
6 INPUT"ZAEHLER:";Z
7 INPUT"NENNER:";N
8 PRINT Z;" / ";N;" = ";
9 IF Z=0 OR N=0 THEN 6
```

```
10 X=Z
11 Y=N
12 R=Y-INT(Y/X)*X
13 IF R=0 THEN 17
14 Y=X
15 X=R
16 GOTO 12
17 WAIT 500
18 D=Z/N
19 PRINT Z/X;" / "; N/X
20 PRINT"GEKUERZT MIT:";X
21 PRINT"DEZIMALBRUCH:"
22 PRINT D
23 INPUT"NEUE RECHNUNG:";N$
24 IF N$="J" THEN 1
25 PRINT "TSCHUESS!!"
26 END
```

Vorschlag:

Versuchen Sie doch einmal die Ausgabe des Dezimalbruches so zu gestalten, daß Sie vorher die Anzahl der Stellen hinter dem Komma bestimmen können.

2.1.2 Wie alt ist Tante Frieda?

Oder: Vom Lösen linearer Gleichungssysteme

Die Aufgabe: Drei Schotten treffen sich auf der Landstraße und beginnen zu feilschen.

Sagt Will zu John: Ich geb dir 6 Schweine für ein Pferd, dann hast du doppelt soviel Tiere in deiner Herde wie ich in meiner.

Darauf sagt Jim zu Will: Ich geb dir 14 Schafe für 1 Pferd, dann hast du dreimal soviel Tiere wie ich.

Do not sale !

Nun sagte John zu Jim: Ich geb dir 4 Kühe für 1 Pferd, dann hast du sechsmal soviel Tiere wie ich.

Wieviel Tiere sind in jeder der drei Herden???

Welcher Schüler würde angesichts solcher Aufgabenstellung nicht lieber die Frage nach Tante Friedas Alter hören, zumal die meisten keine Tante Frieda haben und so die Frage leicht beantworten können. Aber die Fragestellung lautet hier nun mal anders.

Glücklicherweise gibt es jetzt ein Programm im SPCS, welches die Lösung solcher Aufgaben erheblich vereinfacht. Das Aufstellen der notwendigen Gleichungen kann der Computer natürlich nicht übernehmen.

Bei unserem Problem entstehen folgende Gleichungen:

$$\begin{array}{ll} 2(X - 5) = Z + 5 & - \text{Will hat } X \text{ Tiere} \\ 3(Z - 13) = X + 13 & - \text{John hat } Y \text{ Tiere} \\ 6(Y - 3) = Z + 3 & - \text{Jim hat } Z \text{ Tiere} \end{array}$$

Die Gleichungen müssen wie folgt umgeformt werden:

$$\begin{array}{ll} 2X - 10 = Y + 5 & - \quad 2X - 1Y + 0Z = 15 \\ 3Z - 39 = X + 13 & - \quad -1X + 0Y + 3Z = 52 \\ 6Y - 18 = Z + 3 & - \quad 0X + 6Y - 1Z = 21 \end{array}$$

Daraus entsteht folgende Koeffizientenmatrix:

$$\begin{array}{cccc} 2 & -1 & 0 & 15 \\ -1 & 0 & 3 & 51 \\ 0 & 6 & -1 & 21 \end{array}$$

Diese Matrix muß nun zeilenweise eingegeben werden; also:

Do not sale !

2,-1,0,15,-1,0,3,52,0,-1,21

Das Lösen von linearen Gleichungssystemen (bis 3 Gleichungen mit 3 Unbekannten) erfolgt in diesem Programm nach der Regel von Sarrus.

Die Variablenliste entfällt bei diesem Programm, da sie zu umfangreich wäre und die Erläuterungen den geplanten Rahmen sprengen würden.

```
1 CLEAR:WAIT 150
2 PRINT"FRIEDA"
3 DIM A(2,5)
4 DIM H(2)
5 DIM D(3)
6 FOR I=0 TO 2
7 FOR J=0 TO 2
8 INPUT A(I,J)
9 NEXT J
10 INPUT A(I,5)
11 NEXT I
12 FOR N=0 TO 3
13 IF N=0 THEN 19
14 FOR I=0 TO 2
15 IF N>1 THEN LET A(I,N-2)=H(I)
16 H(I)=A(I,N-1)
17 A(I,N-1)=A(I,5)
18 NEXT I
19 FOR I=0 TO 2
20 FOR J=0 TO 1
21 A(I,J+3)=A(I,J)
22 NEXT J
23 NEXT I
24 D=0
25 FOR L=0 TO 5
26 P=1
27 FOR I=0 TO 2
28 IF L<3 THEN 31
```

Do not sale !

```
29 J=L-I-1
30 GOTO 32
31 J=I+L
32 P=A(I,J)*P
33 NEXT I
34 IF L<3 THEN 37
35 D=D-P
36 GOTO 38
37 D=D+P
38 NEXT L
39 D(N)=D
40 ON N GOTO 44,45,46
41 PRINT "D= ";D(0)
42 IF D(0)=0 THEN PRINT"KEINE EIND LSG?"
43 GOTO 47
44 PRINT"D(X)=";D(1):GOTO 47
45 PRINT"D(Y)=";D(2):GOTO 47
46 PRINT"D(Z)=";D(3):GOTO 47
47 NEXT N
48 IF D(0)=0 AND D(1)=0 AND D(2)=0 AND D(3)=0 THEN 51
49 IF D(0)<>0 THEN 52
50 PRINT"KEIN LOESUNG !":END
51 PRINT"MEHRERE LOESUNGEN":END
52 PRINT"X= ";D(1)/D(0)
53 PRINT"Y= ";D(2)/D(0)
54 PRINT"Z= ";D(3)/D(0)
55 END
```

Wer möchte, kann das Programm aber auch auf das Berechnen von 4 Gleichungen mit 4 Unbekannten erweitern. Über die Entwicklung zur letzten Zeile muß man dann für jede Determinante 4 (3x3) Determinanten errechnen. Insgesamt wären das 20 (3x3) Determinanten. Der Zeitaufwand würde sich verfünffachen. Es kann sich also lohnen, sich die Koeffizientenmatrix vorher genau anzuschauen, um durch Zeilentausch möglichst viele 0 in die letzte Zeile zu bekommen.

2.1.3 Körper

Die Berechnung von Körpern ist eine langwierige Angelegenheit, vor allem, wenn man nicht jederzeit die notwendige Formelsammlung zur Hand hat. Doch zum Glück gibt es heute Pocket Computer und die passende Programmsammlung dazu.

Mit Hilfe dieses Programmes können Sie ohne Probleme den Rauminhalt und die Oberfläche von Körpern berechnen. Würfel, Quader, Kegel, Zylinder und Kugel sind dabei die Körper, die für die Berechnung mit diesem Programm ausgewählt wurden.

Direkt nach der Anfangsmeldung gibt der Computer das Hauptmenü auf dem Display aus. Dabei können Sie eine der folgenden 5 Optionen auswählen:

1. Würfel
2. Quader
3. Kegel
4. Zylinder
5. Kugel

Nachdem Sie einen Menüpunkt ausgewählt haben, verlangt der Rechner nach den Informationen, die er zur Berechnung des jeweiligen Körpers benötigt. Dann gibt er das Volumen und die Oberfläche des zu berechnenden Körpers an. Das Programm ist modulmäßig aufgebaut und erlaubt Ihnen, Ihre eigenen Berechnungen in das Programm ohne große Mühe einzubinden. Diese Möglichkeit ist natürlich vor allem für die größeren der PC-Reihe interessant, da der Speicherplatz eine erheblich größere Anzahl verschiedener Berechnungen zuläßt.

Für die Besitzer des PC 1350 bietet sich eine weitere Möglichkeit der Verbesserung an. Das Graphikdisplay ermöglicht eine zusätzliche graphische Darstellung der zu berechnenden Körper.

Variablenliste:

B: Breite
H: Höhe
I: Menüpunkt
K: Kantenlänge
T: Tiefe
O: Oberfläche
R: Radius
V: Volumen

```
1 CLEAR
2 WAIT 150
3 PRINT"BERECHNUNG VON"
4 PRINT"RAUMINHALT UND"
5 PRINT"OBERFLAECHE !"
6 PRINT"WELCHER KOERPER?"
7 PRINT"1=WUERFEL"
8 PRINT"2=QUADER"
9 PRINT"3=KEGEL"
10 PRINT"4=ZYLINDER"
11 PRINT"5=KUGEL"
12 INPUT I
13 IF I<1 OR I>5 THEN PRINT"FALSCH EINGABE":GOTO 12
14 ON I GOSUB 16,20,26,31,36
15 GOTO 48
16 INPUT"KANTENLAENGE:";K
17 V=K^3
18 O=6*K^2
19 RETURN
20 GOSUB 40
21 GOSUB 42
22 GOSUB 44
23 V=H*B*T
24 O=2*(H*B+H*T+B*T)
25 RETURN
26 GOSUB 40
27 GOSUB 46
28 V=1/3*PI*R^2*H
```

Do not sale !


```
29 O=PI*R*(R+(R^2+H^2)^0.5)
30 RETURN
31 GOSUB 40
32 GOSUB 46
33 V=PI*R^2*H
34 O=2*PI*R*(R+H)
35 RETURN
36 GOSUB 46
37 V=4/3*PI*R^3
38 O=4*PI*R^2
39 RETURN
40 INPUT"HOEHE ";H
41 RETURN
42 INPUT"BREITE ";B
43 RETURN
44 INPUT"TIEFE ";T
45 RETURN
46 INPUT"RADIUS ";R
47 RETURN
48 WAIT 300
49 PRINT"VOLUMEN:";V
51 PRINT"OBERFL.:";O
52 INPUT"NEUE RECHNUNG:";A$
53 IF LEFT$(A$,1)="J" THEN 1
54 END
```

2.1.4 ggT und kgV

Im Mathematikunterricht der Klasse 6 werden alle Schüler mit der Berechnung vom größten gemeinsamen Teiler und vom kleinsten gemeinsamen Vielfachen geplagt. Anwendung finden diese Werte im Kürzen von Brüchen (ggT von Zähler und Nenner) und bei der Addition bzw. Subtraktion von Brüchen (kgV der beiden Nenner).

Am günstigsten ist es, den sogenannten euklidischen Algorithmus anzuwenden, der die Nützlichkeit des Distributivgesetzes

(Verteilungsgesetzes) demonstriert. In der Grundschule lernt man bereits das Dividieren mit Rest, d.h. für beliebige ganze Zahlen a und b gilt:

$$a = x \cdot b + R(\text{est}) \quad \text{wobei } x \text{ und } R \text{ ebenfalls ganze Zahlen sind.}$$

Ist nun die ganze Zahl c der ggT (a, b), so muß c wegen der Gültigkeit des Distributivgesetzes auch Teiler des Restes sein, so daß man nur noch den ggT von b und R suchen muß.

Wird R bei wiederholter Anwendung 0, so ist der ggT gefunden. Das kgV (a, b) ist nun $a \cdot b / \text{ggT}(ab)$.

Variablenliste:

A\$:	Antwort
W\$:	Zwischenspeicher
A:	Wert 1
B:	Wert 2
R:	Rest
V:	Kleinste gemeinsames Vielfaches
X:	Zwischenspeicher von A
Y:	Größter gemeinsamer Teiler
Z:	Speicher für Zwischenergebnis

```
1 CLEAR2 WAIT 300
3 PRINT"GGT UND KGV"
4 PRINT"IM SPCS"
5 INPUT"WERT 1:";A
6 INPUT"WERT 2:";B
7 X=A
8 Y=B
9 Z=INT(X/Y)
10 R=X-Y*Z
11 IF R=0 THEN 15
12 X=Y
13 Y=R
14 GOTO 9
```

```
15 V=A*B/Y
16 PRINT"WERT 1:";A
17 PRINT"WERT 2:";B
18 PRINT"DER GGT IST"
19 PRINT Y
20 PRINT"DAS KGV IST"
21 PRINT V
22 INPUT"NEUE RECHNUNG";A$
23 W$=LEFT$(A$,1)
24 IF W$="J" THEN 1
25 PRINT"SEE YOU LATER.."
26 END
```

Der Vorschlag zu diesem Programm ist mehr allgemeiner Natur, denn für diese Problemstellung dürfte die oben gelistete Lösung recht passend sein. Aber es gibt noch eine Menge anderer kleiner Probleme im schulischen Bereich.

2.1.5 Vokabelpauken

Will man eine Fremdsprache erlernen, dann muß man wohl oder übel Vokabeln lernen. Da dies nicht jedem leicht von der Hand geht, hilft dieses Programm beim Lernen. Wenn man das Programm zum ersten Mal starten will, dann muß man RUN 5 eingeben. Beim jedem darauffolgenden Mal muß man nur DEF A eingeben. Nachdem sich das Programm vorgestellt hat, bietet das Menü die Dienste des Programms an:

- E - für das Eingeben von Vokabeln.
- A - für das Anschauen von Vokabeln.
- T - für das Testen der Kenntnisse.
- X - für das Beenden des Programms.

Die im Programm gespeicherten Vokabeln werden auf Karten abgelegt, die von 0 bis 20 durchnummeriert sind. Will man eine Vokabel eingeben, dann wählt man zunächst mit E das Eingeben von Vokabeln an. Nun müssen Sie angeben, wieviele Vokabeln Sie eingeben wollen. Das Programm verlangt danach nach der

Do not sale !

Nummer der Karte, auf der die Vokabel abgelegt werden soll. Geben Sie nun die Nummer 0 an, damit zuerst die erste Karte beschrieben wird. (Natürlich hätte man auch jede andere Karte nehmen können.) Nun erscheint eine 1->, was bedeutet, daß die Vokabel in der ersten Sprache eingegeben werden soll. Leider können nur Vokabeln eingegeben werden, die nicht mehr als 16 Buchstaben haben. Nachdem man die Vokabel in der ersten Sprache eingegeben hat, erscheint eine 2->, was bedeutet, daß die entsprechende Vokabel in der zweiten Sprache eingegeben werden soll. Wenn Sie alle Vokabeln eingegeben haben, dann kehrt das Programm zum Menü zurück.

Wenn Sie sich nun die Vokabeln noch einmal ansehen wollen, dann müssen Sie nur den Menüpunkt A anwählen. Das Programm will dann natürlich wissen, welche Karte Sie sich ansehen wollen. Nachdem Sie sich die entsprechende Karte angesehen haben, können Sie sich weitere Karten ansehen, oder zum Menü zurückkehren.

Testen können Sie Ihr fortschreitendes Wissen mit dem Menüpunkt T. Sie können selbst bestimmen, wieviele Vokabeln Sie der Rechner abfragen soll. Nachdem er Sie alle Vokabeln abgefragt hat, teilt der Rechner Ihnen mit, wieviele Fehler Sie gemacht haben.

Ich hoffe, daß Sie mit diesem Programm Ihre Kenntnisse in den Fremdsprachen weiter verbessern können.

Achtung: Da das Programm ab dem zweiten Mal mit DEF A gestartet wird, werden die eingetippen Vokabeln nicht gelöscht.

Variablenliste:

A\$():	Vokabeln in der ersten Sprache
B\$():	Vokabeln in der zweiten Sprache
C\$():	Antwort beim Test
D\$:	Menüauswahl
N\$:	Antwort

I: Anzahl Karten
 Q: Schleifenvariable
 W: Anzahl der Fehler
 X: Kartennummer

```

1 "A"
2 WAIT 150
3 PRINT"VOKABELPAUKEN"
4 PRINT">> IM SPCS <<":GOTO 8
5 DIM A$(20)
6 DIM B$(20)
7 DIM C$(1)
8 INPUT"E-A-T-X";D$
9 IF D$="E" THEN 15
10 IF D$="A" THEN 28
11 IF D$="T" THEN 38
12 IF D$="X" THEN 53
13 PRINT"FALSCH EINGABE!"
14 WAIT 150:GOTO 8
15 PRINT"EINGABE !!"
16 INPUT"ANZAHL EING:";I
17 FOR Q=1 TO I
18 INPUT"WELCHE KARTE:";X
19 IF X<0 OR X>20 THEN 29
20 IF A$(X)="" THEN 24
21 PRINT"SCHON BELEGT !"
22 INPUT"NEU SCHREIBEN:";N$
23 IF LEFT$(N$,1)<>"J" THEN 18
24 INPUT"1->";A$(X)
25 INPUT"2->";B$(X)
26 NEXT Q
27 GOTO 8
28 PRINT"AUSGABE !!"
29 INPUT"WELCHE KARTE:";I
30 IF I<0 OR I>20 THEN 29
31 PRINT"1->"
32 PRINT A$(I)
33 PRINT"2->"
34 PRINT B$(I)

```

Do not sale !

```
35 INPUT"WEITERE: ";N$
36 IF N$="J" THEN 29
37 GOTO 8
38 PRINT"TEST !!":W=0
39 INPUT"ANZAHL VOK. ";I
40 FOR Q=1 TO I
41 X=RND 20
42 IF A$(X)="" THEN 41
43 PRINT"1->"
44 PRINT A$(X)
45 INPUT"2-> ";C$(1)
46 IF B$(X)=C$(1) THEN 49
47 W=W+1
48 PRINT"FEHLER !"
49 NEXT Q
50 PRINT"VON ";I;" VOKABELN"
51 PRINT W;" FALSCH"
52 GOTO 8
53 PRINT"PROGRAMM BEENDEN"
54 INPUT"ENDE (J/N) ";N$
55 IF N$="J" THEN 59
56 IF N$="N" THEN 8
57 PRINT"FALSCH EINGABE!"
58 GOTO 8
59 PRINT"TSCHUESS !!"
60 END
```

Ich bin zwar der Meinung, daß dieses Programm alle Voraussetzungen eines Vokabelprogramms erfüllt, aber trotzdem lassen sich noch viele Verbesserungen durchführen. Beim Test werden die Vokabeln durch Zufallszahlen bestimmt. Diese Methode erscheint zwar auf dem ersten Blick ideal, aber beim zweiten Hinsehen werden Sie bemerken, daß die Möglichkeit besteht, daß eine Vokabel mehrmals nacheinander abgefragt wird. Diese Tatsache liegt sicher nicht im Dienste des Lernerfolges. Man müßte eine Routine einschieben, die sich merkt, welche Vokabeln schon abgefragt wurden.

2.1.6 Pythagoras

Der Satz des Pythagoras ermöglicht die Berechnung der Hypothenuse in einem rechtwinkligen Dreieck. Dazu benötigt man die Länge der Ankathete und der Gegenkathete. Die Berechnung an sich ist denkbar einfach.

Die Wurzel aus der Summe des Quadrates der Ankathete und des Quadrates der Gegenkathete ergibt die Länge der Hypothenuse.

Variablenliste:

N\$:	Neue Rechnung Ja, Nein
A:	Ankathete
B:	Gegenkathete
C:	Hypothenuse
D:	Quadrat der Ankathete
E:	Quadrat der Gegenkathete
F:	Summe der Quadrate

```
1 CLEAR
2 WAIT 250
3 PRINT"DREIECK"
4 PRINT"IM SPCS"
5 INPUT"ANKATHETE";A
6 INPUT"GEGENKATH.";B
7 D=A*A
8 E=B*B
9 F=D+E
10 C=SQR F
11 PRINT"DIE LAENGE DER"
12 PRINT"HYPOTHENUSE"
13 PRINT"BETRAEGT:";C
14 INPUT"NEUE RECHNUNG:";N$
15 IF LEFT$(N$,1)="J" THEN 1
16 PRINT"TSCHUESS !"
17 END
```

Do not sale !

2.1.7 Mathe-Trainer

Kopfrechnen schwach! Jeder, der sich diese Bemerkung schon einmal anhören mußte, sollte nicht lange überlegen, sondern dieses Programm abtippen. Denn schwache Leistungen im Kopfrechnen ergeben sich nicht aus mangelnder Intelligenz, sondern vielmehr aus Faulheit. Natürlich kann man Kopfrechnen auch ohne Programm üben, aber es gibt mindestens drei gute Gründe es mit einem Programm zu tun.

1. Man muß sich seine Aufgaben nicht auch noch vorher ausdenken.
2. Es macht viel mehr Spaß, mit dem Computer zu arbeiten.
3. Man läßt auch die Ergebnisse vom Computer überprüfen.

Ich glaube, jeder dieser drei Gründe wäre allein schon ausreichend, um das Arbeiten mit einem Programm zu rechtfertigen.

Das Programm bietet das Rechentraining in allen 4 Grundrechenarten an. Nachdem man sich für eine entschieden hat, kann man auch noch den Schwierigkeitsgrad der Aufgaben festlegen. Es gibt die beiden Schwierigkeitsgrade "leicht" und "schwer". So klar und deutlich sind diese Unterscheidungsmerkmale anhören, so deutlich sind sie auch. Bei dem ersten Schwierigkeitsgrad werden sehr kleine Zahlen generiert. Beim zweiten Grad werden die Zahlen um ein Vielfaches höher. Wer alle Aufgaben in allen Rechenarten bei schwieriger Aufgabenstellung im Kopf lösen kann, der kommt im Alltag mit diesen Fertigkeiten sicher gut zurecht.

Variablenliste:

N\$:	Antwort
A:	Menüauswahl
B:	Schwierigkeitsgrad
P:	Erste Zahl

Q: Zweite Zahl
R: Ergebnis aus P und Q
X: Zahlenbereich für P und Q

```
1 CLEAR
2 WAIT 150
3 PRINT"MATHE-TRAINER"
4 PRINT">> IM SPCS <<"
5 PRINT"1.ADDIEREN"
6 PRINT"2.SUBTRAH."
7 PRINT"3.MULTIPL."
8 PRINT"4.DIVID."
9 PRINT"5.ENDE"
10 INPUT"1,2,3,4,5 :";A
11 IF A>0 AND A<6 THEN 15
12 IF A=5 THEN PRINT"BYE BYE!":END
13 PRINT"FEHLER !!"
14 GOTO 5
15 PRINT"1-LEICHT,2-SCHWER"
16 INPUT B
17 IF B<1 OR B>2 THEN 15
18 IF B=1 THEN LET X=15
19 IF B=2 THEN LET X=5000
20 ON A GOTO 22,32,43,53
21 GOTO 5
22 P=RND X
23 Q=RND X
24 PRINT P;"+";Q;"= ?"
25 INPUT R
26 IF R=P+Q THEN PRINT"BRAVO!"
27 IF R<>P+Q THEN PRINT"FAST.."
28 PRINT P;"+";Q;"=";P+Q
29 INPUT"NOCHMAL:";N$
30 IF LEFT$(N$,1)="J" THEN 22
31 GOTO 5
32 P=RND X
33 Q=RND X
34 IF P-Q<0 THEN 32
35 PRINT Q;"-";P;"= ?"
```

Do not sale !

```
36 INPUT R
37 IF R=P-Q THEN PRINT"BRAVO!"
38 IF R<>P-Q THEN PRINT"FAST.."
39 PRINT P;"-";Q;"=";P-Q
40 INPUT"NOCHMAL: ";N$
41 IF LEFT$(N$,1)="J" THEN 32
42 GOTO 5
43 P=RND X
44 Q=RND X
45 PRINT P;"*";Q;"= ?"
46 INPUT R
47 IF R=P*Q THEN PRINT"BRAVO!"
48 IF R<>P*Q THEN PRINT"FAST.."
49 PRINT P;"*";Q;"=";P*Q
50 INPUT"NOCHMAL: ";N$
51 IF LEFT$(N$,1)="J" THEN 43
52 GOTO 5
53 P=RND 10*X
54 Q=RND 10*X
55 IF P/Q<>INT(P/Q) THEN 53
56 PRINT P;" / ";Q;"=";P/Q
57 INPUT R
58 IF R=P/Q THEN PRINT"BRAVO!"
59 IF R<>P/Q THEN PRINT"FAST.."
60 PRINT P;" / ";Q;"=";P/Q
61 INPUT"NOCHMAL: ";N$
62 IF LEFT$(N$,1)="J" THEN 53
63 GOTO 5
```

Die Zahlenbereiche, aus denen die Werte für Q und P gewählt werden, können von Ihnen sehr leicht beeinflußt werden. In den Zeilen 18 und 19 werden die Werte von X festgelegt. Dabei sind diese Werte für jeden Schwierigkeitsgrad einzeln abzuändern. Eine andere Möglichkeit der Beeinflussung der Schwierigkeiten vom Programm aus wäre ein erweitertes Menü für die Schwierigkeitsgrade. Man könnte dabei z.B. von sehr leicht bis zur Kopfschmerzstufe differenzieren. Die Unterteilung ist Ihnen

natürlich völlig überlassen. Eine solche Erweiterung macht praktisch keine Veränderung in der Programmstruktur nötig und ist somit sehr einfach durchzuführen.

2.1.8 Ohmsches Gesetz

Eines der bekanntesten und zugleich eines der wichtigsten physikalischen Gesetze in der Elektrizitätslehre ist das Ohmsche Gesetz, mit dessen Hilfe man die Beziehungen zwischen Stromstärke, Spannung und Widerstand darstellen und berechnen kann.

Ohmsches Gesetz:

In einem elektrischen Stromkreis ist die Stromstärke (I) der Spannung (U) direkt und dem Widerstand (R) umgekehrt proportional.

Also gilt $I=U/R$

Kennt man die Beziehungen zwischen diesen 3 Größen, so kann man immer aus zwei dieser Größen die dritte berechnen. Genau das leistet dieses Programm. Im Menü können Sie wählen, welche der 3 Größen Sie berechnen wollen. Haben Sie einen Menüpunkt ausgewählt, so müssen Sie natürlich die beiden Ihnen bekannten Werte angeben. Dabei müssen Sie die Werte in folgenden Einheiten angeben:

Widerstand in Ohm	$V/A = \text{kg} * \text{m}^2 / \text{s}^3 * \text{A}^2$
Spannung in Volt	$W/A = \text{kg} * \text{m}^2 / \text{s}^3 * \text{A}$
Stromstärke in Ampere	A

Danach gibt Ihnen das Programm den gesuchten Wert auf dem Display an. Die Hauptaufgabe dieses Programms ist natürlich die Berechnung von Werten, aber Sie können mit Hilfe dieses Programms auch Ihre eigenen Ergebnisse überprüfen.

Do not sale !

Variablenliste:

N\$: Antwort
W\$: Zwischenspeicher
A: Menüauswahl
I: Stromstärke
R: Widerstand
U: Spannung

```
1 CLEAR
2 WAIT 150
3 PRINT"OHMSCHES GESETZ"
4 PRINT">> IM SPCS <<"
5 PRINT"1.WIDERSTAND"
6 PRINT"2.SPANNUNG"
7 PRINT"3.STROMSTAERKE"
8 PRINT"4.MENUE NOCHM."
9 INPUT"1,2,3 ODER 4 :";A
10 IF A>0 AND A<5 THEN 13
11 PRINT"FEHLER !!!"
12 GOTO 9
13 IF A=2 THEN 23
14 IF A=3 THEN 30
15 IF A=4 THEN 5
16 INPUT"SPANNUNG:";U
17 INPUT"STROMST.:";I
18 R=U/I
19 PRINT U;" VOLT"
20 PRINT I;" AMP="
21 PRINT R;" OHM"
22 GOTO 36
23 INPUT"WIDERST.:";R
24 INPUT"STROMST.:";I
25 U=R*I
26 PRINT R;" OHM"
27 PRINT I;" AMP="
28 PRINT U;" VOLT"
29 GOTO 36
30 INPUT"WIDERST.:";R
```

Do not sale !

```
31 INPUT"SPANNUNG:";U
32 I=U/R
33 PRINT R;" OHM"
34 PRINT U;" VOLT="
35 PRINT I;" AMP"
36 INPUT"NEUE RECHNUNG:";N$
37 W$=LEFT$(N$,1)
38 IF W$="J" THEN 1
39 IF W$="N" THEN 42
40 PRINT"FALSCHE EINGABE!"
41 GOTO 36
42 PRINT"TSCHUESS!"
43 END
```

Die in der Programmbeschreibung bereits angesprochene Überprüfung Ihrer eigenen Rechenergebnisse können Sie auch in das vorliegende Programm einbinden. Dabei sollte die Möglichkeit gegeben werden, alle Werte einschließlich des Ergebnisses anzugeben, und der Computer stellt dann fest, ob Sie richtig gerechnet haben.

2.1.9 Pfuschomat

Dieses Programm ist das Ergebnis der konsequenten Weiterentwicklung der weltweiten Pfusch- und Schummeltechnologie. Mit dem Pfuschomat können Sie sich optimal auf Prüfungen jeglicher Art vorbereiten.

Der Pfuschomat bietet Ihnen in der Grundversion die Möglichkeit, 50 kurze Merksätze und Formeln in Ihrem PC abzulegen und bei Bedarf abzurufen. Dabei kann jeder Merksatz 16 Zeichen beinhalten.

Nachdem man das Programm mit DEF A gestartet hat, bringt der Rechner einen ins Menue, welches 3 Wahlmöglichkeiten bietet.

Do not sale !

1. Eingabe von Merksätzen.
2. Ausgabe von Merksätzen.
3. Löschen des Programms.
4. Menü wiederholen.
5. Beenden des Programms

Die einzelnen Merksätze sind als Blöcke bezeichnet. Wenn man also Merksätze eingeben will, so erkundigt sich der Rechner zunächst nach der Blocknummer. Dann kann man seine Schummelkartei verändern, wie man will. Ein besonderes Bonbon beim Pfuschat ist die Lehrersicherung. Es soll ja inzwischen Lehrer geben, die wissen, daß Sharp PC keine Taschenrechner, sondern Computer sind. Für diese Exemplare hält der Rechner eine kleine Überraschung bereit. Versucht er ein vermeindliches Programm mit RUN zu starten, so meldet sich der Rechner sogleich mit der lapidaren Erklärung "MEMORY IS FREE" ab. Sogleich wird der Lehrer annehmen, daß der Rechner zwar programmierbar, aber eben nicht programmiert ist. Wenn man das Programm auch noch vorher mit einem Passwort gesichert hat, kann er es auch nicht listen.

Viel Glück beim Pfuschen!

Variablenliste:

A\$:	Textblöcke
E\$:	Tastaturabfrage
N\$:	Antwort
W\$:	Zwischenspeicher
B:	Menuewahl
I:	Blocknummer

```
1 PRINT"MEMORY IS FREE":END
2 "A":CLEAR:WAIT 150
3 PRINT">PFUSCHAT<"
4 PRINT">> IM SPCS<<"
5 DIM A$(50)
6 PRINT"1.EINGABE"
```

Do not sale !

```
7 PRINT"2.AUSGABE"  
8 PRINT"3.LOESCHEN"  
9 PRINT"4.MENUE NOCHM."  
10 PRINT"5.ENDE"  
11 INPUT"1,2,3,4,5 :";B  
12 IF B>0 AND B<6 THEN 15  
13 PRINT"FALSCHE EINGABE!"  
14 GOTO 11  
15 IF B=4 THEN 6  
16 IF B=3 THEN 36  
17 IF B=2 THEN 29  
18 IF B=5 THEN 46  
19 PRINT"EINGABE !!"  
20 INPUT"BLOCK:";I  
21 IF I<0 OR I>50 THEN 20  
22 INPUT A$(I)  
23 PRINT"BLOCK:";I  
24 PRINT A$(I)  
25 INPUT"O.K. :";N$  
26 W$=LEFT$(N$,1)  
27 IF W$<>"J" THEN 20  
28 GOTO 6  
29 PRINT"AUSGABE !!"  
30 INPUT"BLOCK:";I  
31 IF I<0 OR I>50 THEN 30  
32 PRINT A$(I)  
33 E$=INKEY$  
34 IF E$="X" THEN 6  
35 GOTO 32  
36 PRINT"LOESCHEN !!"  
37 PRINT"WILLST DU"  
38 PRINT"ALLES LOESCHEN?"  
39 E$=INKEY$  
40 IF E$="" THEN 39  
41 IF E$="J" THEN 43  
42 GOTO 6  
43 INPUT"SICHER:";E$  
44 IF E$<>"J" THEN 6  
45 NEW  
46 PRINT"BEENDEN !!"
```

Do not sale !

```
47 INPUT"ENDE :";NS
48 WS=LEFT$(NS,1)
49 IF WS="J" THEN 53
50 IF WS="N" THEN 6
51 PRINT"FALSCH EINGABE!"
52 GOTO 46
53 PRINT"HAE, HAE !!!"
54 END
```

Wem die 16 Zeichen pro Block zuwenig sind, der kann mehrere Blocks zu einem Block zusammenfassen. Bei der Realisierung einer solchen Veränderung sollte man sich allerdings auch einen Weg zum durchscrollen des Textes überlegen. Es gibt sicher noch viele Möglichkeiten, ein Pfschprogramm zu verbessern und zu verändern.

2.2 Programme für den Beruf

Die Programme in diesem Themenbereich sind für Anwender gedacht, die Ihren Sharp PC auch im Beruf einsetzen wollen. Auch hier können die 9 Programme in diesem Kapitel nur kleine Bereiche des beruflichen Einsatzbereiches ansprechen. Aber die Programme sollen nicht nur fertige Software für den Benutzer darstellen, sondern besonders in diesem Bereich Hinweise und Anregungen für eigene Programme geben. Denn die besten Programme sind die, die für eine spezielle Problematik erstellt worden sind. Ich bin sicher, daß die Programme in diesem Kapitel wichtige Hinweise zur Erstellung von Programmen für den beruflichen Alltag geben können.

1. Abschreibungen

Alle drei Abschreibungsarten lassen sich mit diesem Programm durchführen. Sowohl lineare und degressive, als auch digitale Abschreibungen lassen sich einfach berechnen. Das Programm ist voll menügesteuert.

2. Calc

Eine Tabellenkalkulation in kleinem Maßstab, stellt das Programm Calc dar. Es soll als Grundlage für größere Kalkulationen dienen. Natürlich kann man das Programm auch in der vorliegenden Form nutzen.

3. Yards1

Yards1 ist das erste von drei Programmen, die sich mit der Problematik der Umrechnung nicht metrischer Einheiten beschäftigen. In diesem Programm werden yards, feet und inches in inches umgerechnet. Selbstverständlich bietet das Programm auch die umgekehrte Berechnung an.

4. Yards2

Mit Yards2 wird die Umrechnungsproblematik logisch weitergeführt. Hier wird nun aus yards, feet und inches yards-Dezimale errechnet. Auch hier ist natürlich der umgekehrte Rechenweg Bestandteil des Programms, der im Menü angeboten wird.

5. Yards3

Drittes und letztes Programm der Yards-Programme. Hier wird nun die Umrechnung von Dezimal-Yards in Metern vollzogen. Wie auch in den anderen beiden Programmen, so ist auch hier die umgekehrte Rechnung möglich.

6. Optimal

Nach Umrechnungen in verschiedenen Einheiten wird nun eine Hilfestellung zur Lösung eines der größten betriebswirtschaftlichen Probleme angeboten. Die Optimale Bestellmenge für einen Betrieb wird mit dem Programm Optimal angeboten.

Do not sale !

7. Meilen

Meilen pro Stunde, Kilometer pro Stunde und Knoten werden menügesteuert umgerechnet. Man gibt einen Wert in einer der 3 Einheiten an und die entsprechenden Werte in den anderen Einheiten werden sofort ausgegeben.

8. Mittelwert

Den Mittelwert aus einer Reihe von Daten errechnet das achte Programm dieses Kapitels. Zusätzlich werden die Extremwerte der Datenreihe angegeben. Auch dieses Programm soll als Grundlage für spezielle Mittelwertprogramme dienen.

9. Währ2

Das Umrechnen einer ausländischen Währungseinheit in eine zweite fremde Sorte ist eine umständliche Angelegenheit. Schnell und ohne Umstände, mit Rücksicht auf die Besonderheiten des Währungssystems, erledigt das Programm diese Aufgabe.

2.2.1 Abschreibungen

Dieses Programm ist quasi der große Bruder des Restwert-Programms. Während das Restwertprogramm lediglich auf der degressiven Abschreibung beruhte, bietet dieses Programm die Auswahl unter den verschiedenen Abschreibungsmöglichkeiten. Sie können im Menü zwischen linearer, degressiver und digitaler Abschreibung wählen. Der Rechner verlangt dann nach dem Neuwert und der Lebensdauer der abzuschreibenden Objekte. Wenn Sie die degressive Abschreibung gewählt haben, dann müssen Sie natürlich auch den Prozentsatz angeben. Nachdem alle Daten angegeben worden sind, können Sie den Wert des Objektes im jeweiligen Abschreibungsjahr abfragen.

Variablenliste:

M\$:	Abfrage
B(N):	Wert des Objektes
D:	Zwischenspeicher
I:	Schleifenvariable
J:	Jahr
N:	Lebensdauer
P:	Prozentsatz
R:	Zählvariable
S:	Anschaffungswert
U:	Zählvariable
W:	Menüwert

```
1 CLEAR
2 WAIT 150
3 PRINT"ABSCHREIBUNG"
4 PRINT"ANSCHAFFUNGSWERT"
5 INPUT S
6 INPUT"LEBENSDAUER:";N
7 IF N>10 THEN DIM B(N)
8 B(0)=S
9 PRINT"WIE WOLLEN SIE"
10 PRINT"ABSCHREIBEN..."
11 PRINT"LINEAR(1)"
12 PRINT"DEGRESSIV(2)"
13 PRINT"DIGITAL(3)"
14 INPUT" 1,2 ODER 3 :";W
15 IF W<1 OR W>3 THEN PRINT"FEHLER !":GOTO 14
16 ON W GOSUB 18,24,30
17 GOTO 2
18 PRINT"LINEAR..."
19 FOR I=1 TO N
20 D=S/N
21 B(I)=B(I-1)-D
22 NEXT I
23 GOTO 39
24 PRINT"DEGRESSIV..."
```

Do not sale !

```
25 INPUT"PROZENTSATZ: ";P
26 FOR I=1 TO N
27 B(I)=B(I-1)-(B(I-1)*P/100)
28 NEXT I
29 GOTO 39
30 PRINT"DIGITAL..."
31 FOR I=1 TO N
32 R=R+I
33 NEXT I
34 D=S/R
35 FOR I=1 TO N
36 U=U+1
37 B(I)=B(I-1)-(R*(N-(U-1)))
38 NEXT I
39 PRINT"ERGEBNIS:"
40 GOSUB 48
41 INPUT"WELCHES JAHR: ";J
42 WAIT 300
43 PRINT J;" JAHR: ";B(J)
44 PRINT"NOCH EIN JAHR"
45 INPUT M$
46 IF LEFT$(M$,1)="J" THEN 41
47 GOTO 52
48 FOR I=1 TO N
49 B(I)=(INT(B(I)*100+0.5))/100
50 NEXT I
51 RETURN
52 INPUT"NEUE RECHNUNG: ";M$
53 IF LEFT$(M$,1)="J" THEN 1
54 END
```

2.2.2 Calc

Quickcalc ist, man höre und staune, eine Tabellenkalkulation für den SPCS. Dieses Programm benutzt ein 5 x 5 Feld, um Werte zu verarbeiten. Man hat die Möglichkeit zeilenorientiert zu addieren und man kann auch spaltenweise addieren. Die Handhabung dieses Programms ist einfach und daher auch praktikabel.

Do not sale !

Die folgenden Funktionen können genutzt werden:

- A - Eine Zeile nach unten
- E - Eine Spalte nach rechts
- G - Gesamtwerte nach Modus
- I - Moduseingabe
- Q - Eine Zeile nach oben
- W - Eine Spalte nach links
- X - Eingabe eines Wertes

Zu den Funktionen I, G und X muß noch ein erklärendes Wort gesagt werden. Die Moduseingabe (I) legt durch S oder Z fest, ob spaltenweise oder zeilenweise addiert werden soll. Die Gesamtausgabe (G) gibt entsprechend der Modusart die Gesamtsummen je Spalte bzw. Zeile aus. Die Eingabe eines Wertes in ein Feld wird durch den X-Befehl eingeleitet. Der Rechner fragt nach dem Wert und nach der Eingabe des Wertes, gefolgt von ENTER, muß man den Wert noch durch J bestätigen.

Variablenliste:

- C\$: Modusanzeige
- D\$: Tastaturabfrage
- E\$: Tastaturabfrage
- A: Koordinate 1
- B: Koordinate 2
- G(5): Feld für Summen
- I: Schleifenvariable
- P: Schleifenvariable
- U: Schaltvariable
- Z(4,4): Feld für Werte

Do not sale !

```
1 CLEAR
2 WAIT 150
3 PRINT"QUICKCALC"
4 PRINT"IM SPCS"
5 DIM G(5)
6 DIM Z(4,4)
7 A=0:B=0
8 WAIT 50:PRINT Z(A,B);";";A;";";B;C$:WAIT 150
9 D$=INKEY$
10 IF D$="" THEN 9
11 IF D$="Q" THEN LET A=A-1
12 IF D$="A" THEN LET A=A+1
13 IF D$="W" THEN LET B=B-1
14 IF D$="E" THEN LET B=B+1
15 IF D$="I" THEN GOSUB 23
16 IF D$="X" THEN GOSUB 35
17 IF D$="G" THEN GOSUB 42
18 IF A>4 THEN LET A=4
19 IF A<0 THEN LET A=0
20 IF B>4 THEN LET B=4
21 IF B<0 THEN LET B=0
22 GOTO 8
23 INPUT"MODUS:";C$
24 IF C$<>"Z" AND C$<>"S" THEN 23
25 IF C$="Z" THEN U=1
26 IF C$="S" THEN U=2
27 FOR I=0 TO 4: G(I)=0:NEXT I
28 FOR I=0 TO 4
29 FOR P=0 TO 4
30 IF U=1 THEN LET G(I)=G(I)+Z(I,P)
31 IF U=2 THEN LET G(I)=G(I)+Z(P,I)
32 NEXT P
33 NEXT I
34 RETURN
35 PRINT A;";";B;
36 INPUT"WERT:";Z(A,B)
37 PRINT A;";";B;"=";Z(A,B);"O.K."
38 E$=INKEY$
39 IF E$="" THEN 38
40 IF E$="J" THEN RETURN
```

Do not sale !

```
41 GOTO 35
42 PRINT"GESAMT:"
43 FOR I=0 TO 4
44 IF CS="Z" THEN PRINT"ZEILE";I
45 IF CS="S" THEN PRINT"SPALTE";I
46 PRINT G(I)
47 WAIT 150
48 NEXT I
49 RETURN
```

Wer Lust und Zeit hat, der kann noch andere Rechenarten in das Programm aufnehmen. Da die Variablenlänge allerdings auf 7 Zeichen beschränkt ist, muß man bei den anderen Rechenarten aufpassen.

2.2.3 Yards und Inches

Dieses Programm fällt in die Kategorie der Umrechnungsprogramme von englischen bzw. amerikanischen Maßen und Gewichten. Da den englischen Maßen kein metrisches System zugrunde liegt, gestaltet sich die Umrechnung sehr schwierig. Es gibt folgende Einheiten, die auch in den USA Gültigkeit haben:

1 mile = 1760 yards
1 yard = 3 feet
1 foot = 12 inches
1 inch = 12 lines

Die Schreibung 13.3.4 bedeutet 13 yards 2 feet 9 inches. Die Umrechnung von yards und feet in inches erfolgt, indem man die yards mit 3 multipliziert und dann die feet mit 12. Die Umrechnung in umgekehrter Richtung ist natürlich nicht weniger umständlich. Zunächst muß man die inches durch 12 dividieren, zum Ergebnis die angegebenen feet addieren und diese Summe durch 3 teilen.

Da diese Umrechnung sehr umständlich ist, eignet sie sich diese Problematik für ein Programm, das diese Berechnung für den

Do not sale !

Benutzer vornimmt. Das Programm bietet zu Beginn des Programms die Wahl zwischen der ersten und der zweiten Umrechnung. Bei der Eingabe ist das Programm komfortabel gestaltet. Man kann die yards, feet und inches in der gewohnten Form angeben (also: y.f.i).

Natürlich deckt dieses Programm noch nicht die gesamte Problematik der Umrechnung ab. Für die noch offenen Probleme folgen Programme auf den nächsten Seiten.

Variablenliste:

C\$:	Zwischenspeicher
D\$():	Teile von Y\$
N\$:	Antwort
W\$:	Zwischenspeicher
Z\$:	Ergebnis
A:	Menüauswahl
F:	Zwischenspeicher
E():	Zahlenwerte der Eingabe
I:	Schleifenvariable
Q:	Zwischenspeicher
R:	Ergebnis
S:	Zwischenspeicher
Y:	Zwischenspeicher
X:	Zählvariable

```
1 CLEAR
2 WAIT 150
3 DIM D$(2)
4 DIM E(2)
5 PRINT"YARDS + INCHES"
6 PRINT">> IM SPCS <<"
7 PRINT"1.YDS->INS"
8 PRINT"2.INS->YDS"
9 PRINT"3.MENUE NOCHMAL"
10 INPUT"1,2 ODER 3:";A
```

Do not sale !


```
11 IF A=3 THEN 7
12 IF A=2 THEN 28
13 INPUT"YARDS:";Y$
14 FOR I=1 TO LEN Y$
15 C$=MID$(Y$,I,1)
16 IF C$="." THEN LET X=X+1:GOTO 18
17 D$(X)=D$(X)+C$
18 NEXT I
19 FOR I=0 TO 2
20 E(I)=VAL D$(I)
21 NEXT I
22 R=R+E(0)*36
23 R=R+E(1)*12
24 R=R+E(2)
25 PRINT Y$;" YARDS="
26 PRINT R;" INCHES"
27 GOTO 39
28 INPUT"INCHES:";R
29 F=INT(R/12)
30 Q=R-(F*12)
31 Y=INT(F/3)
32 S=F-(Y*3)
33 F=S
34 Z$=STR$ Y+"."
35 Z$=Z$+STR$ F+"."
36 Z$=Z$+STR$ Q
37 PRINT R;" INCHES="
38 PRINT Z$;" YDS"
39 INPUT"NEUE RECHNUG:";N$
40 W$=LEFT$(N$,1)
41 IF W$="J" THEN 1
42 IF W$="N" THEN 45
43 PRINT"FALSCH EINGABE!"
44 GOTO 39
45 PRINT"SO LONG"
46 END
```

Als Anregung bleibt in diesem Fall nur der Vorschlag, die 3 Programme, die sich mit der Yardsumrechnung beschäftigen, in

Do not sale !

einem zusammenzufassen. Dies ist natürlich nur bei Modellen mit einer Speicherkapazität von über 3000 Bytes möglich.

2.2.4 Yards - Dezimal

Nachdem im vorherigen Programm inches in yards und feet umgerechnet worden sind, erfolgt in diesem Programm die Umrechnung von feet und inches in yards-Dezimalzahlen. Auch hier entpuppt sich die Umrechnung im nicht metrischen System als problematisch. Natürlich bietet auch dieses Programm die umgekehrte Rechnung an.

Die Umrechnung von feet und inches erfolgt, indem man die inches durch 12 dividiert, zum Ergebnis die angegebenen feet addiert und die Summe durch 3 dividiert. Bei der umgekehrten Rechnung muß man die yards-Dezimalzahlen mit 3 multiplizieren und dann die feet-Dezimalzahlen mit 12. Für beide Berechnungen gibt es auch einen anderen Lösungsweg. Dabei macht man sich folgende Tatsachen zunutze:

$$\begin{aligned}1 \text{ foot} &= 1/3 \text{ yard} = 0,333 \text{ yard} \\1 \text{ inch} &= 1/36 \text{ yard} = 0,028 \text{ yard} \\1 \text{ inch} &= 1/12 \text{ foot}\end{aligned}$$

Will man also feet und inches in yards-Dezimalzahlen umwandeln, so kann man auch die feet mit 0,333 und die inches mit 0,028 multiplizieren und die Ergebnisse addieren. Bei der umgekehrten Rechnung verhält es sich dann entsprechend anders. Man dividiert die yards-Dezimalen durch 0,333 und den Rest durch 0,028.

Variablenliste:

A:	Menüauswahl
E():	Werte aus DS()
F:	Zwischenspeicher
G:	Zwischenspeicher
H:	Ergebnis

Do not sale !

Q: Zwischenspeicher
 R: Zwischenspeicher
 S: Zwischenspeicher
 X: Zählvariable
 Y: yards
 C\$: Teilstrings aus Y\$
 D\$(*i*): Zwischenspeicher
 N\$: Antwort
 W\$: Zwischenspeicher
 Y\$: yards

```

1 CLEAR
2 WAIT 150
3 DIM D$(2)
4 DIM E(2)
5 PRINT">>YARDS-DEZ<<"
6 PRINT">> IM SPCS <<"
7 PRINT"1.F+I->YDS"
8 PRINT"2.YDS->F+I"
9 PRINT"3.MENUE NOCHMAL"
10 INPUT"1,2 ODER 3:";A
11 IF A=3 THEN 7
12 IF A=2 THEN 28
13 INPUT"YARDS:";Y$
14 FOR I=1 TO LEN Y$
15 C$=MID$(Y$,I,1)
16 IF C$="." THEN LET X=X+1:GOTO 18
17 D$(X)=D$(X)+C$
18 NEXT I
19 FOR I=0 TO 2
20 E(I)=VAL D$(I)
21 NEXT I
22 F=E(2)/12
23 G=(F+E(1))/3
24 H=E(0)+G
25 USING"###.####"
26 PRINT H" YDS"
27 GOTO 39
28 INPUT"YDS:";Y
  
```

Do not sale !

```
29 G=INT Y
30 Q=Y-G
31 H=INT(Q*3)
32 R=(Q*3)-H
33 S=INT(R*12)
34 T$=STR$ G+"."
35 T$=T$+STR$ H+"."
36 T$=T$+STR$ S
37 PRINT Y;" YDS="
38 PRINT T$;" YDS"
39 INPUT"NEUE RECHNUNG: ";N$
40 W$=LEFT$(N$,1)
41 IF W$="J" THEN 1
42 IF W$="N" THEN 45
43 PRINT"FALSCH EINGABE!"
44 GOTO 39
45 PRINT"ADE !"
46 END
```

Wenn Sie Spaß am Programmieren haben, dann sollten Sie versuchen, das Programm so umzuschreiben, daß die zweite Lösungsmöglichkeit verwandt wird. Vorab möchte ich bemerken, daß die zweite Möglichkeit die einfachere ist.

2.2.5 Yards und Meter

Das dritte und letzte Programm, das sich mit der Umrechnung von yards beschäftigt, rundet das Bild ab. Nachdem zuvor yards und feet in inches und yards und feet in yards-Dezimale umgerechnet worden sind, werden im nachfolgend aufgelisteten Programm yards-Dezimale in Meter umgerechnet. Wie bei den beiden Programmen zuvor, so ist auch hier die umgekehrte Rechnung möglich. Da für die eigentliche Umrechnung von yards in Meter die yardszahl in dezimalen vorliegen muß, ist eine Routine zur Umrechnung von yards, feet und inches in yards-Dezimale im Programm implementiert. Sie können also die Werte wie gewohnt angeben (y.f.i). Bei der Umrechnung von yards in Meter liegen folgende Beziehungen zugrunde:

Do not sale !

1 yard = 0,9144 m

12 yards = 11 m

Obwohl die zweite Relation nicht 100%ig exakt ist, wird sie häufig in der Praxis angewandt. Da uns aber ein Computer bei der Berechnung hilft, können wir die erste und genauere Relation zugrundelegen. In der Berechnung wird zunächst die yards-Dezimale berechnet und mit 0,9144 multipliziert. Bei der Umkehrrechnung verhält es sich genauso, nur wird hier durch 0,9144 dividiert.

Variablenliste:

A:	Menüauswahl
E():	Werte aus D\$()
F:	Zwischenspeicher
G:	Zwischenspeicher
H:	Zwischenspeicher
I:	Schleifenvariable
R:	Zwischenspeicher
S:	Zwischenspeicher
X:	Meter
C\$:	Zwischenspeicher
D\$():	Teilstrings aus Y\$
N\$:	Antwort
T\$:	Zwischenspeicher
W\$:	Zwischenspeicher
Y\$:	yards

```
1 CLEAR
2 WAIT 150
3 DIM D$(2)
4 DIM E(2)
5 PRINT"YARDS<->METER"
6 PRINT"1. YDS -> METER"
7 PRINT"2. METER -> YDS"
8 PRINT"3.MENUE NOCHMAL"
```

Do not sale !

```
9 INPUT A
10 IF A>0 AND A<4 THEN 13
11 PRINT"FEHLER !!"
12 GOTO 5
13 IF A=3 THEN 6
14 IF A=2 THEN 31
15 INPUT"YARDS:";Y$
16 FOR I=1 TO LEN Y$
17 C$=MID$(Y$,I,1)
18 IF C$="." THEN LET X=X+1:GOTO 20
19 D$(X)=D$(X)+C$
20 NEXT I
21 FOR I=0 TO 2
22 E(I)=VAL D$(I)
23 NEXT I
24 F=E(2)/12
25 G=(F+E(1))/3
26 H=E(0)+G
27 X=H*0.9144
28 PRINT Y$;" YDS="
29 PRINT X;" M"
30 GOTO 43
31 INPUT"METER:";M
32 C=M/0.9144
33 G=INT C
34 Q=C-G
35 H=INT(Q*3)
36 R=(Q*3)-H
37 S=INT(R*12)
38 T$=STR$ G+". "
39 T$=T$+STR$ H+". "
40 T$=T$+STR$ S
41 PRINT M;" METER="
42 PRINT T$;" YDS"
43 INPUT"NEUE RECHNUNG:";N$
44 W$=LEFT$(N$,1)
45 IF W$="J" THEN 1
46 IF W$="N" THEN 49
```

Do not sale !

```
47 PRINT"FALSCH EINGABE!"  
48 GOTO 43  
49 PRINT"ADE !"  
50 END
```

Wenn die im Programm angewandte Methode genauer sein mag, so kann es doch nichts schaden, wenn Sie versuchen, die andere Methode in das Programm einzubinden. So könnte der Anwender wählen, welche Methode der Berechnung er anwenden will.

2.2.6 Optimal

Eines der größten Probleme der Betriebswirtschaft liegt in der Bestimmung der optimalen Losgröße. Diese Erkenntnis ist nicht neu. Neu ist aber, daß es ein Programm im SPCS gibt, mit dessen Hilfe man dieser Problematik Herr werden kann.

Damit dieses Programm einwandfrei arbeiten kann, muß man zunächst die 4 Ausgangsgrößen der Berechnung angeben. Bei den 4 Ausgangsgrößen handelt es sich um:

1. Den durchschnittlichen Jahresverbrauch
2. Die Höhe der Rüstkosten
3. Die Stückkosten des Produktes
4. Den durchschnittlichen Lagerzinssatz

Nachdem man diese 4 Werte angegeben hat, berechnet das Programm die optimale Losgröße für das zu produzierende Produkt.

Definition:

Die optimale Losgröße ist die Produktionsmenge (Menge der gleichartig zu bearbeitenden Teile), bei der die Summe aus Lager- und Rüstkosten am geringsten ist.

Variablenliste:

J:	Jahresbedarf
K:	Stückkosten
L:	Lagerzins
P:	Zwischenspeicher
Q:	Zwischenspeicher
R:	Rüstkosten
X:	Zwischenspeicher
Y:	Optimale Losgröße

```
1 CLEAR
2 WAIT 150
3 PRINT">>OPTIMAL<<"
4 PRINT">> IM SPCS <<"
5 PRINT"WIE HOCH IST"
6 INPUT"DER JAHRESBEDARF: ";J
7 PRINT"HOEHE DER "
8 INPUT"RUESTKOSTEN: ";R
9 PRINT"HERSTELLKOSTEN"
10 INPUT"PRO STUECK: ";K
11 PRINT"HOEHE DES "
12 INPUT"LAGERZINSES: ";L
13 Q=200*J
14 P=Q/L
15 X=P*R
16 Y=(X/K)^0.5
17 PRINT"ERGEBNIS !!"
18 PRINT"JAHRESBEDARF: ";J
19 PRINT"RUESTKOSTEN: ";R
20 PRINT"STUECKKOSTEN: ";K
21 PRINT"LAGERZINS: ";L
22 PRINT"DIE OPTIMALE"
23 PRINT"LOSGROESSE= ";Y
24 PRINT"WERTE ERNEUT"
25 INPUT"ZEIGEN ";M$
26 W$=LEFT$(M$,1)
27 IF W$="J" THEN 18
```



```

28 INPUT"NEUE RECHNUNG:";M$
29 W$=LEFT$(M$,1)
30 IF W$="J" THEN 1
31 PRINT"SEE YOU LATER.."
32 END

```

Wer sich für diese Problematik näher interessiert, der sollte versuchen die Berechnung des Lagerzinses und die Höhe der Stückkosten mit in das Programm aufzunehmen.

2.2.6 mph - km/h - kn

Mit diesem Programm können Sie Stundenkilometer in Meilen pro Stunde und Knoten umrechnen. Diese Umrechnung läßt sich von jeder dieser Maßeinheit aus machen, d.h. Sie können einen Wert entweder in Stundenkilometern, in Meilen pro Stunde oder in Knoten angeben und das Programm berechnet die jeweils entsprechenden Werte in den anderen beiden Einheiten. Die Umrechnung an sich gestaltet sich nicht als besonders schwierig, wenn man folgende Relationen zugrunde legt:

1 Meile = 1.609 Kilometer
1 Knoten = 1 Seemeile pro Stunde
1 Seemeile = 1.852 Kilometer

Aus diesen Beziehungen ergeben sich aus die anderen Beziehungen untereinander.

Variablenliste:

L\$:	Antwort
A:	Menüauswahl
K:	KM/H
M:	MPH
N:	KN

Do not sale !

```
1 CLEAR
2 WAIT 150
3 PRINT"MPH-KM/H-KN"
4 PRINT"1.KM/H -> XXX"
5 PRINT"2. XXX -> KM/H"
6 PRINT"3.MENUE WIEDERH."
7 INPUT"1,2 ODER 3 :";A
8 IF A>0 AND A<4 THEN 11
9 PRINT"FALSCH EINGABE!"
10 GOTO 7
11 IF A=3 THEN 4
12 IF A=2 THEN 24
13 INPUT"KM/H : ";K
14 M=K/1.609
15 N=K/1.852
16 PRINT K;" KM/H="
17 PRINT M;" MPH="
18 PRINT N;" KN"
19 INPUT"ERGEBNIS NOCHM.";L$
20 IF LEFT$(L$,1)="J" THEN 16
21 IF LEFT$(L$,1)="N" THEN 50
22 PRINT"FALSCH EINGABE!"
23 GOTO 19
24 PRINT"1.MPH"
25 PRINT"2.KNOTEN"
26 INPUT"1 ODER 2 :";A
27 IF A=2 THEN 39
28 INPUT"MPH:";M
29 K=M*1.609
30 N=M/1.15102548
31 PRINT M;" MPH="
32 PRINT K;" KM/H="
33 PRINT N;" KN"
34 INPUT"ERGEBNIS NOCHM.";L$
35 IF LEFT$(L$,1)="J" THEN 31
36 IF LEFT$(L$,1)="N" THEN 50
37 PRINT"FALSCH EINGABE!"
38 GOTO 34
39 INPUT"KNOTEN:";N
40 K=N*1.852
```

Do not sale !

```
41 M=N/0.868790497
42 PRINT N;" KN="
43 PRINT K;" KM/H="
44 PRINT M;" MPH"
45 INPUT"ERGEBNIS NOCHM.";L$
46 IF LEFT$(L$,1)="J" THEN 42
47 IF LEFT$(L$,1)="N" THEN 50
48 PRINT"FALSCH EINGABE!"
49 GOTO 45
50 INPUT"NEUE RECHNUG.";L$
51 IF LEFT$(L$,1)="J" THEN 1
52 IF LEFT$(L$,1)="N" THEN 55
53 PRINT"FALSCH EINGABE!"
54 GOTO 50
55 PRINT"CIAO !!"
56 END
```

Eine interessante Erweiterung dieses Programms wäre sicherlich die zusätzliche Berechnung von yards pro Stunde. Da spielt natürlich wieder die nicht metrische Gestalt dieser Einheit eine wichtige Rolle. Trotzdem sollten Sie versuchen, diese Berechnung in das Programm einzubinden. Sollten Sie Probleme bei der Auswertung der yards-Werte haben, dann sehen Sie sich einfach die entsprechenden Routinen bei den yards-Programmen an.

2.2.7 Mittelwert

Mittelwertberechnungen fallen sehr häufig an, doch in den meisten Fällen ist es mit dem Mittelwert allein nicht getan. Oft werden auch die extremen Werte verlangt. Das Programm Mittelwert, welches nachfolgend aufgelistet ist, berechnet den Mittelwert aus einer Reihe von Daten. Die Menge der zu berechnenden Daten können Sie selbst zu Beginn des Programms festlegen. Zum Mittelwert gibt das Programm auch noch den größten und den kleinsten Wert der Daten an. Natürlich ist das Programm nicht für die Praxis geeignet, dafür ist es viel zu unspezialisiert. Doch eine sinnvolle Spezialisierung dieses Programms kann nur der vornehmen, der mit der Problematik in

der Praxis vertraut ist. Somit ist dieses Programm nur eine Grundlage für Ihr eigenes spezielles Mittelwertprogramm, mit dem Sie Ihre Probleme in der Praxis besser lösen können.

Variablenliste:

A\$:	Überschrift
N\$:	Antwort
W\$:	Zwischenspeicher
B:	Größter Wert
C:	Kleinster Wert
D:	Mittelwert
I:	Schleifenvariable
X:	Anzahl der Werte
Y:	Summe aller Werte
Z():	Werte

```
1 CLEAR
2 WAIT 150
3 PRINT">MITTELWERT<"
4 PRINT">> IM SPCS <<"
5 INPUT"WIEVIEL WERTE:";X
6 IF X<50 THEN 9
7 PRINT"ZU VIELE !!"
8 GOTO 5
9 DIM Z(X)
10 PRINT"UEBERSCHRIFT ?"
11 INPUT A$
12 FOR I=1 TO X
13 PRINT I;:INPUT" ";Z(I)
14 IF Z(I)>C THEN LET C=Z(I)
15 NEXT I
16 FOR I=1 TO X
17 Y=Y+Z(I)
18 IF B<Z(I) THEN LET B=Z(I)
19 IF C>Z(I) THEN LET C=Z(I)
20 NEXT I
21 D=Y/X
```

Do not sale !

```
22 PRINT"SUMME: ";Y
23 PRINT"MITTELWERT VON"
24 PRINT A$
25 PRINT "=";D
26 PRINT"GROESSTER WERT="
27 PRINT B
28 PRINT"KLEINSTER WERT="
29 PRINT C
30 INPUT"NEUE RECHNUNG: ";N$
31 W$=LEFT$(N$,1)
32 IF W$="J" THEN 1
33 IF W$="N" THEN 36
34 PRINT"FALSCH EINGABE!"
35 GOTO 30
36 PRINT"CIAO !!"
37 END
```

2.2.8 Währungen

Jeder, der schon einmal im Ausland Urlaub gemacht hat, weiß wie schwierig sich das Umrechnen von fremden Währungseinheiten in der Praxis gestaltet. Auch mit den kleinen Tabellen, die das Umrechnen erleichtern sollen, ist es nicht ohne weiteres möglich, den genauen Betrag zu ermitteln.

Den genauen DM-Betrag in Fremdwährung und auch die umgekehrte Rechnung kann man mit Hilfe dieses Programm ermitteln. Dabei nimmt das Programm automatisch auf die Besonderheiten bei Pfund, Dollar und Lire Rücksicht. Bei diesem Programm zahlt sich wieder einmal die echte Portabilität der Sharp PC aus. Mit anderen Computern wäre eine solche Anwendung in der Praxis kaum möglich, da die meisten Portabels über 8 KG wiegen.

Nachdem man das Programm gestartet hat, stellt einem der Rechner zur Wahl, ob man 1. ausländische Währung in DM umrechnen will oder 2. DM in ausländische Währung umrechnen will.

Do not sale !

Dann muß man den Namen der Geldsorte eingeben. Dabei sollte man sich auf die vorgegebenen Abkürzungen beschränken. Besonders sollten Sie auf die Abkürzungen von Dollar (\$), Pfund (PFD) und Lire (LIT) achten, da diese Abkürzungen für die Berechnung relevant sind. Die Abkürzung für Pfund muß PFD heißen, da das Pfundzeichen nicht im Zeichensatz der Sharp PC enthalten ist.

Wenn man den Namen der Geldsorte eingegeben hat, dann muß man als nächstes den Betrag angeben. Die dritte Eingabe bezieht sich auf den Wechselkurs der jeweiligen Geldsorte. Nach diesen 3 Eingaben gibt Ihnen das Programm den Betrag der anderen Geldsorte an.

Variablenliste:

E\$:	Sortenname
F\$:	DM
G\$:	Zwischenspeicher
N\$:	Antwort
W\$:	Zwischenspeicher
A:	Welche Berechnung
B:	Betrag
C:	Zwischenspeicher
D:	Ergebnis
K:	Kurs
U:	Länge des Sortennamens
Z:	Divident

```
1 CLEAR
2 WAIT 150
3 Z=100
4 PRINT">>WAEHRUNGEN<<"
5 PRINT">> IM SPCS <<"
6 PRINT"1.AUSL. IN DM"
7 PRINT"2.DM IN AUSL."
8 PRINT"3.MENUE NOCHMAL"
9 PRINT"1,2 oder 3"
```

```
10 INPUT A
11 IF A<1 OR A>3 THEN 6
12 IF A=3 THEN 6
13 F$="DM"
14 PRINT"NAME DER SORTE"
15 INPUT E$
16 U=LEN E$
17 IF U>1 AND U<4 THEN 19
18 PRINT"FEHLER !!":GOTO 14
19 IF E$="LIT" THEN LET Z=1000
20 IF E$="PFD" THEN LET Z=1
21 IF E$="$" THEN LET Z=1
22 INPUT"BETRAG: ";B
23 INPUT"KURS: ";K
24 IF A=2 THEN 27
25 D=B*K/Z
26 GOTO 28
27 D=B*Z/K
28 IF A=2 THEN LET C=D:D=B:D=C
29 IF A=2 THEN LET G$=E$:E$=F$:F$=G$
30 PRINT B;" ";E$;"="
31 PRINT D;" ";F$
32 INPUT"NEUE RECHNUNG:";N$
33 W$=LEFT$(N$,1)
34 IF W$="J" THEN 2
35 IF W$="N" THEN 37
36 GOTO 32
37 PRINT"BYE, BYE !"
38 END
```

Wenn Ihnen dieses Programm noch nicht genug Komfort bieten sollte, dann sollten Sie versuchen, das Programm Währungen 2 in dieses Programm mit einzubinden. Dies konnte an dieser Stelle aus Gründen des Speicherplatzes nicht geschehen, da die kleinen Rechner im SPCS nicht über ausreichenden Speicherplatz verfügen.

Do not sale !

2.2.9 Währungen 2

Nachdem Sie mit dem ersten Währungsprogramm DM in Fremdwährung umrechnen konnten, gibt Ihnen dieses Programm die Möglichkeit, mit zwei Fremdwährungen zu rechnen. Sie geben die Namen und die Kurse der beiden Sorten an und der Rechner berechnet aus dem Betrag der einen Sorte den Betrag der entsprechenden anderen Sorte. Natürlich nimmt auch dieses Programm Rücksicht auf die Besonderheiten der italienischen Lire, des englischen Pfunds und des amerikanischen Dollars. Dabei ist es bei Umrechnungen dieser Währungen unerlässlich, auf die richtigen Abkürzungen zu achten.

Pfund = PFD

Lire = LIT

Dollar = \$

So leicht das Programm einem die Umrechnung macht, die Entscheidung, ob man Brief- oder Geldkurs angibt, kann einem der Rechner nicht abnehmen. Deshalb sollten Sie sich vorher noch einmal genau über diese Problematik informieren.

Variablenliste:

A\$:	Name der ersten Sorte
B\$:	Name der zweiten Sorte
N\$:	Antwort
W\$:	Zwischenspeicher
E:	Kurs 1
F:	Kurs 2
G:	Betrag
Y:	Teiler
Z:	Teiler


```

1 CLEAR
2 WAIT 150
3 Y=100:Z=100
4 PRINT">WAEHRUNGEN2<"
5 PRINT">> IM SPCS <<"
6 INPUT"1. SORTE: ";A$
7 IF LEN A$<1 OR LEN A$>3 THEN 11
8 INPUT"2. SORTE: ";B$
9 IF LEN B$<1 OR LEN B$>3 THEN 11
10 GOTO 13
11 PRINT"FEHLER !"
12 GOTO 6
13 PRINT"BETRAG ";A$
14 INPUT G
15 PRINT"KURS ";A$
16 INPUT E
17 PRINT"KURS ";B$
18 INPUT F
19 IF A$="PFD" THEN LET Y=1
20 IF A$="$" THEN LET Y=1
21 IF A$="LIT" THEN LET Y=1000
22 IF B$="PFD" THEN LET Z=1
23 IF B$="$" THEN LET Z=1
24 IF B$="LIT" THEN LET Z=1000
25 D=(G*E*Z)/(Y*F)
26 PRINT G;" ";A$;"="
27 PRINT D;" ";B$
28 INPUT "NEUE RECHNUNG: ";N$
29 W$=LEFT$(N$,1)
30 IF W$="J" THEN 1
31 IF W$="N" THEN 34
32 PRINT"FALSCH EINGABE"
33 GOTO 28
34 PRINT"ADE !"
35 END

```

Da sich die Kurse über einen kurzen Zeitraum kaum verändern, könnte es sich lohnen, die Kurse der wichtigsten Währungen fest im Speicher abzulegen.

Do not sale !

Sollten Sie dieses Programm bei einer Urlaubsreise anwenden wollen, so sollten Sie die notwendigen Kurse fest in Ihr Programm einbauen. So kann man noch schneller mit dem Programm arbeiten. Es wären dann nur noch die Namen der Sorten und der Betrag anzugeben, alles andere erledigt sich dann von selbst.

Speziell für eine Urlaubsreise könnte es sich auch lohnen, ein völlig neues Programm zu entwickeln, welches nur die entsprechende Währung verarbeiten kann. Der Vorteil eines solchen Programms liegt auf der Hand. Die Verarbeitung geht wesentlich schneller vonstatten.

2.3 Programme für den Haushalt

Interessante Anwendungen im alltäglichen Bereich werden in diesem Kapitel angeboten. Insgesamt 8 Programme, die häufig, ja sogar täglich gebraucht werden, sind hier aufgelistet. Die Problematik für Programme, die im Haushalt genutzt werden sollen, liegt in dem Wert für die Praxis. Die Programme müssen so gestaltet sein, daß sie tatsächlich genutzt werden können und genutzt werden. Denn was nützen die schönsten Programme, wenn Ihnen jeder Praxisbezug fehlt?

1. Übergewicht??

Mit diesem Programm und einer Personenwaage, können Sie einen Überblick über die Gewichtssituation Ihres Körpers gewinnen. Viele Menschen glauben, daß sie über- oder untergewichtig sind, ohne eine Richtlinie zu haben. Mit diesem Programm haben Sie eine Linie für Ihre Linie.

2. Sprit

Der Benzinverbrauch des Wagens ist ein fester Kostenfaktor im Haushaltsbudget. Mit diesem Programm wissen Sie über den genauen Verbrauch Ihres Fahrzeugs Bescheid.

3. Takt

Beim telefonieren verliert man sehr schnell den Überblick über die verbrauchten Einheiten. Mit diesem Programm neben dem Telefon wissen sie immer genau, wieviel von Ihrem Geld gerade durch die Kupferleitung wandert.

4. Restwert

Den ungefähren Wiederverkaufswert von Haushaltsgegenständen berechnet das Restwertprogramm in Anlehnung an die degressive Abschreibung.

5. Währungen

Für die Umrechnung von Währungseinheiten im Urlaub oder bei ähnlichen Gelegenheiten sorgt dieses Programm. Es rechnet nach Wahl DM in Fremdwährung und Fremdwährung in DM-Beträge um.

6. Lotto 1 und 2

Hier handelt es sich um zwei eigenständige Programme. Das erste erleichtert Ihnen die Zahlenwahl beim Spiel 6 aus 49, während das zweite die 6 aus 45 zieht.

7. Zinseszins

In diesem Programm wird die Zinseszinsberechnung vom Rechner durchgeführt. Sie müssen lediglich die Werte eingeben und das Programm errechnet das Ergebnis.

8. Darlehen

Die Höhe eines aufzunehmenden Darlehens wird von diesem Programm berechnet. Wenn Sie ein Darlehen aufnehmen wollen, dann sollten Sie mit diesem Programm die für Sie vertretbare Höhe des Darlehens berechnen.

2.3.1 Übergewicht??

Wenn Sie zu den Menschen zählen, die sich über diese Frage keine Gedanken machen oder machen brauchen, dann können Sie sich glücklich schätzen. Viele Menschen, die keine Gewichtsprobleme haben, unterschätzen die Problematik für die Betroffenen. Wie (ge)wichtig das richtige Gewicht genommen wird, zeigt sich, wenn man bedenkt, daß ganze Industrien davon leben. Ich will Ihnen an dieser Stelle kein Programm vorstellen, das Ihnen Ihren Diätplan für die nächste Woche ausdruckt, sondern vielmehr möchte ich Ihnen die Möglichkeit geben, Ihr Gewicht richtig einzuschätzen. Die falsche Einschätzung des eigenen Gewichtes kann eine teure Angelegenheit werden. Deshalb sagt Ihnen das nachfolgend aufgelistete Programm, ob Sie Übergewicht, Untergewicht oder Normalgewicht haben. Zusätzlich gibt Ihnen das Programm Auskunft darüber, um wieviel Sie von Ihrem Normalgewicht abweichen.

Sollte Ihr Gewicht erheblich vom Normalgewicht abweichen, so ist das noch lange kein Grund für einen Systemabsturz. Sollten Sie zu denen gehören, die unter die Kategorie Übergewicht fallen, dann sollten Sie lediglich etwas weniger und dafür etwas gesünder essen. Für diejenigen, die an Untergewicht leiden, habe ich nur einen Rat: Essen, Trinken und Schlafen macht Spaß - versuchen Sie es mal.

Do not sale !

In diesem Sinne wünsche ich Ihnen einen guten Appetit.

Variablenliste:

H\$:	Anrede
M\$:	Antwort
N\$:	Name
R\$:	Normalgewicht
SS:	Geschlecht
WS:	Zwischenspeicher
A:	Alter in Jahren
E:	Normalgewicht
D:	Untergewicht
G:	Körpergewicht in kg
K:	Koerpergröße in cm
N:	Idealgewicht
U:	Übergewicht
Y:	Abweichung von Normal

```
1 CLEAR
2 WAIT 150
3 R$="NORMAL"
4 PRINT"UEBERGEWICHT ?"
5 PRINT">> IM SPCS <<"
6 INPUT"NAME: ";N$
7 PRINT"KOERPER-"
8 INPUT"GROESSE: ";K
9 INPUT"GEWICHT: ";G
10 PRINT"GESCHLECHT"
11 INPUT"M/W: ";S$
12 INPUT"ALTER: ";A
13 IF A<5 OR A>110 THEN 51
14 IF S$="W" THEN 25
15 IF S$="M" THEN 18
16 PRINT"NUR M ODER W"
17 GOTO 10
18 IF K>230 OR K<60 THEN 51
```

Do not sale !

```
19 IF G>K*2 THEN 51
20 E=K-100
21 N=E*0.9
22 U=E*1.2
23 D=E*0.8
24 GOTO 29
25 E=(K-100)*0.9
26 N=E*0.9
27 U=E*1.2
28 D=E*0.9
29 IF G>U THEN LET R$="UEBER"
30 IF G<D THEN LET R$="UNTER"
31 IF G>=N AND G<E THEN LET R$="IDEAL"
32 IF S$="M" THEN LET H$="HERR "
33 IF S$="W" THEN LET H$="FRAU "
34 Y=G-E
35 PRINT H$
36 PRINT N$
37 PRINT"SIE HABEN"
38 PRINT R$;"GEWICHT"
39 PRINT"IHR GEWICHT"
40 PRINT"WEICHT: ";Y;" KG"
41 PRINT"VOM NORMALGEW.AB"
42 INPUT"NEUE RECHNUNG: ";M$
43 W$=LEFT$(M$,1)
44 IF W$="J" THEN 1
45 IF W$="N" THEN 48
46 PRINT"FALSCH EINGABE!"
47 GOTO 42
48 PRINT"GUTEN"
49 PRINT"HUNGER !!"
50 END
51 PRINT"UEBERPRUEFEN"
52 PRINT"SIE IHRE"
53 PRINT"ANGABEN !!"
54 GOTO 6
```

Sollten Sie im Besitz eines etwas größeren PC sein (>5 kBytes), dann sollten Sie eine entsprechende Ernährungsempfehlung in

Do not sale !

das Programm aufnehmen. Diese Empfehlung können Sie natürlich nach eigenem Gutdünken einsetzen. Interessant wäre es, bei Übergewicht Sahnebonbons und bei Untergewicht Magerjoghurt zu verschreiben. Es besteht dann allerdings die Möglichkeit, das Ihr Programm etwas an Glaubwürdigkeit einbüßt.

2.3.2 Sprit

Die Benzinpreise schlagen ja bekanntlich in letzter Zeit die tollsten Kapriolen. Nichtsdestotrotz ist man als Nichtmillionär gezwungen, auf seine Ausgaben zu achten. Mit diesem Programm kann man seinen Benzinverbrauch auf 100 Kilometer berechnen.

Nachdem man das Programm gestartet hat, verlangt der Rechner zuerst nach der Anzahl der gefahrenen Kilometer. Wenn man diesen Wert angegeben hat, dann muß man den Verbrauch, also die verbrauchte Benzinmenge, angeben. Anders als andere Programme dieser Art, verlangt dieses Programm noch die Angabe, ob zugetankt wurde oder nicht. Schlußendlich gibt dieses Programm den Benzinverbrauch auf 100 Kilometer in Litern an.

Variablenliste:

A\$:	Zugetankt?
K:	Anzahl der Kilometer
V:	Verbrauchtes Benzin
Z:	Zugetankte Liter

```
1 CLEAR
2 WAIT 150
3 PRINT"SPRITVERBRAUCH"
4 PRINT" IM SPCS"
5 INPUT"ANZAHL DER KM: ";K
6 INPUT"VERBRAUCH: ";V
7 INPUT"ZUGETANKT: ";A$
8 B$=LEFT$(A$,1)
```

Do not sale !

```
9 IF BS<>"J" THEN 12
10 INPUT"WIEVIEL LITER:";Z
11 V=V+Z
12 D=100*V/K
13 PRINT"VERBRAUCH -100/KM"
14 D=(INT(D*100+0.5))/100
15 PRINT D;"LTR"
16 INPUT"NEUE RECHNUNG:";BS
17 AS=LEFT$(BS,1)
18 IF AS="J" THEN 1
19 PRINT"ENDE"
20 END
```

Jetzt wissen wir, wie hoch unser Benzinverbrauch auf 100 Kilometer ist. Nun wäre es noch interessant, diesen Verbrauch in direkte Beziehung zur aufgewendeten Geldmenge zu sehen. Es wäre ohne großen Aufwand möglich, eine Routine einzuschieben, die nach Angabe des Benzinpreises, die Kosten von 100 gefahrenen Kilometern berechnet. Dann wüßten Sie endlich ganz genau, was Sie der Fahrspaß gekostet hat.

Aber Vorsicht! Vielleicht wäre es Ihnen nachher lieber, wenn Sie es gar nicht so genau wüßten.

2.3.3 Telefon-Takt

Der größte Vorteil eines Pocket Computers ist seine absolute Portabilität. Die sich daraus ergebende Flexibilität läßt sich in der Praxis durch nichts wieder wett machen.

Dieses Programm nutzt die Beweglichkeit voll aus. Wo immer Sie sich auch aufhalten mögen, Sie müssen nur Ihren Sharp aus der Tasche nehmen, das Telefon-Takt Programm starten und schon haben Sie Ihre Telefonkosten immer unter Kontrolle.

Nach dem Programmstart müssen Sie zunächst angeben, ob Sie die Auskunft in Takteinheiten oder in DM haben wollen. Die nächste entscheidende Frage, ist die Frage, ob Sie im Nahbe-

Do not sale !

reich oder im Fernbereich telefonieren wollen. Sollten Sie im Nahbereich telefonieren, so können Sie schon nach diesen zwei Eingaben das Programm starten.

Wenn Sie aber im Fernbereich telefonieren, dann müssen Sie zusätzlich angeben, ob gerade ein Feiertag ist. Diese Frage ist im Hinblick auf die Gebühren sehr wichtig, da es ja an Feiertagen bekanntlich billiger ist. Billiger ist es auch in der Zeit von 18 Uhr bis 6 Uhr. Deshalb müssen Sie die Frage nach der Uhrzeit auch noch beantworten. Die letzte Frage, ist die Frage nach der Anzahl der Sekunden pro Takt. Sollten Sie zu denen gehören, die die entsprechende Tabelle nicht im Kopf haben, dann sollten Sie das Programm um diese Werte erweitern.

Die Liste mit den entsprechenden Werten bekommen Sie in jeder größeren Poststelle.

Variablenliste:

A\$:	Verschiedene Antworten
R\$:	Antwort
W\$:	Zwischenspeicher
I:	Schleifenvariable
J:	Schleifenvariable
K:	Anzahl der Takte
Q:	Zählvariable
P:	Sekunden pro Takt
U:	Uhrzeit
Y:	Entscheidung Takt/DM

```

1 CLEAR
2 WAIT 150
3 PRINT"TAKT"
4 PRINT"IM SPCS"
5 INPUT"TAKT:DM (T/D)";RS
6 W$=LEFT$(RS,1)
7 IF W$="T" THEN LET Y=1
8 IF W$="D" THEN LET Y=2

```

Do not sale !

```
9 INPUT"NAHBEREICH <J/N>";AS
10 W$=LEFT$(A$,1)
11 IF W$<>"J" THEN 22
12 INPUT"IST FEIERTAG:";AS
13 W$=LEFT$(A$,1)
14 IF W$="J" THEN 21
15 INPUT"UHRZEIT:";U
16 INPUT"TAKT:DM (T/D)";R$
17 W$=LEFT$(R$,1)
18 IF W$="T" THEN LET Y=1
19 IF W$="D" THEN LET Y=2
20 IF U<18 AND U>=6 THEN LET P=480:GOTO 27
21 P=720:GOTO 27
22 INPUT"SEC TAKT";P
23 PRINT"S---> START"
24 N$=INKEY$:IF N$="" THEN 24
25 IF N$="S" THEN 24
26 K=1
27 Q=P
28 FOR I=1 TO P
29 FOR J=I TO 750
30 NEXT J
31 Q=Q-1
32 IF Q=0 THEN 34
33 NEXT I
34 K=K+1
35 D=K*0.23
36 IF Y=2 THEN 39
37 PRINT K;" TAKT"
38 GOTO 40
39 PRINT D;" DM"
40 GOTO 27
```

Möglichkeiten dieses Programm zu erweitern, gibt es genug. Da wäre zum Beispiel die Möglichkeit, die Taktlängen in Sekunden mit in das Programm aufzunehmen. Eine andere Erweiterung, wäre eine Routine, die bemerkt, wenn man von 17.45 Uhr bis 18.15 Uhr telefoniert. Die also die Veränderung der Gebühren entsprechend registriert.

Do not sale !

2.3.4 Restwertberechnung

Es ist traurig aber wahr: Nachdem Sie einen Gegenstand gekauft haben, beginnt sein Wert sich auf mehr oder weniger drastische Weise zu verringern. Zwar gibt es auch hier Ausnahmen, aber die bestätigen wieder einmal nur die Regel. Wenn Sie nun wissen wollen, was Ihr Auto oder Ihr Fernseher demnächst noch wert sein wird, dann sollten Sie das folgende Programm schnellstens in Ihren Rechner eintippen. Natürlich kann dieses Programm nur eine Richtlinie sein, denn der Wert von gebrauchten Gegenständen hängt ja von sehr vielen Kriterien ab.

Programmablauf:

Das Programm wird mit DEF A gestartet. Nach dem Titel verlangt das Programm zunächst den Neuwert des zu berechnenden Objektes. Da es sich bei diesem Programm um eine erweiterte Restwertberechnung handelt, verlangt das Programm nun die Eingabe des heutigen Wertes. Der Jetztwert muß nicht 100%ig stimmen, denn er dient nur zur Erhöhung der Ergebnisgenauigkeit. Zum Schluß wird noch die Angabe des Alters benötigt. Die Altersangabe sollte auf ein halbes Jahr genau sein. Nachdem alle Werte eingegeben worden sind, berechnet der Rechner zunächst die Wertminderungsrate für das zu berechnende Objekt. Nun gibt der Rechner den Restwert des Objektes bis zu 8 Jahren an.

Variablenliste:

Z\$:	Antwort
I:	Jahre
J:	Alter in Jahren
N:	Neuwert
M:	Jetztwert
R:	Wertminderungsrate: Restwert

```
1 "A":CLEAR
2 WAIT 150
3 PRINT"RESTWERT"
4 INPUT"NEUWERT:";N
5 INPUT"JETZTWERT:";M
6 PRINT"ALTER IN JAHREN:"
7 INPUT J
8 R=(N-M)
9 R=(R*100/N)/J
10 PRINT"WERTMINDERUNGS-"
11 PRINT"RATE ";R;"%"
12 Q=R/100
13 FOR I=J+1 TO 8
14 R=N*((1-Q)^(I+J))
15 WAIT 300
16 PRINT"RESTWERT IM"
17 PRINT I;"JAHR=";R
18 NEXT I
19 INPUT "NEUE RECHNUNG:";Z$
20 IF Z$="J" THEN 1
21 IF Z$="N" THEN END
22 PRINT"BITTE J ODER N"
23 GOTO 19
```

2.3.5 Währungen

Jeder, der schon einmal im Ausland Urlaub gemacht hat, weiß, wie schwierig sich das Umrechnen von fremden Währungseinheiten in der Praxis gestaltet. Auch mit den kleinen Tabellen, die das Umrechnen erleichtern sollen, ist es nicht ohne weiteres möglich, den genauen Betrag zu ermitteln.

Den genauen DM-Betrag in Fremdwährung und auch die umgekehrte Rechnung, kann man mit Hilfe dieses Programmes ermitteln. Dabei nimmt das Programm automatisch auf die Besonderheiten bei Pfund, Dollar und Lire Rücksicht. Bei diesem Programm zahlt sich wieder einmal die echte Portabilität der Sharp

PC aus. Mit anderen Computern wäre eine solche Anwendung in der Praxis kaum möglich, da die meisten Portabels über 8 KG wiegen.

Nachdem man das Programm gestartet hat, stellt einem der Rechner zur Wahl, ob man 1. ausländische Währung in DM umrechnen will, oder 2. DM in ausländische Währung umrechnen will. Dann muß man den Namen der Geldsorte eingeben. Dabei sollte man sich auf die vorgegebenen Abkürzungen beschränken. Besonders sollten Sie auf die Abkürzungen von Dollar (\$), Pfund (PFD) und Lire (LIT) achten, da diese Abkürzungen für die Berechnung relevant sind. Die Abkürzung für Pfund muß PFD heißen, da daß Pfundzeichen nicht im Zeichensatz der Sharp PC enthalten ist.

Wenn man den Namen der Geldsorte eingegeben hat, dann muß man als nächstes den Betrag angeben. Die dritte Eingabe bezieht sich auf den Wechselkurs der jeweiligen Geldsorte. Nach diesen 3 Eingaben gibt Ihnen das Programm den Betrag der anderen Geldsorte an.

Variablenliste:

E\$:	Sortenname
F\$:	DM
G\$:	Zwischenspeicher
N\$:	Antwort
W\$:	Zwischenspeicher
A:	Welche Berechnung
B:	Betrag
C:	Zwischenspeicher
D:	Ergebnis
K:	Kurs
U:	Länge des Sortennamens
Z:	Divident

Do not sale !

```
1 CLEAR
2 WAIT 150
3 Z=100
4 PRINT">>WAHRUNGEN<<"
5 PRINT">> IM SPCS <<"
6 PRINT"1.AUSL. IN DM"
7 PRINT"2.DM IN AUSL."
8 PRINT"3.MENUE NOCHMAL"
9 PRINT"1,2 oder 3"
10 INPUT A
11 IF A<1 OR A>3 THEN 6
12 IF A=3 THEN 6
13 F$="DM"
14 PRINT"NAME DER SORTE"
15 INPUT E$
16 U=LEN E$
17 IF U>1 AND U<4 THEN 19
18 PRINT"FEHLER !!!":GOTO 14
19 IF E$="LIT" THEN LET Z=1000
20 IF E$="PFD" THEN LET Z=1
21 IF E$="$" THEN LET Z=1
22 INPUT"BETRAG: ";B
23 INPUT"KURS: ";K
24 IF A=2 THEN 27
25 D=B*K/Z
26 GOTO 28
27 D=B*Z/K
28 IF A=2 THEN LET C=D:D=B:D=C
29 IF A=2 THEN LET G$=E$:E$=F$:F$=G$
30 PRINT B;" ";E$;"="
31 PRINT D;" ";F$
32 INPUT"NEUE RECHNUNG: ";N$
33 W$=LEFT$(N$,1)
34 IF W$="J" THEN 2
35 IF W$="N" THEN 37
36 GOTO 32
37 PRINT"BYE, BYE !"
38 END
```

Do not sale !

Wenn Ihnen dieses Programm noch nicht genug Komfort bieten sollte, dann sollten Sie versuchen, das Programm Währungen 2 in dieses Programm mit einzubinden. Dies konnte an dieser Stelle aus Gründen des Speicherplatzes nicht geschehen, da die kleinen Rechner im SPCS nicht über ausreichenden Speicherplatz verfügen.

2.3.6 Lottozahlen

Sollten auch Sie zu den Tippfern gehören, denen die Entscheidung für die richtigen Zahlen schwerfällt, dann sind die beiden nachfolgenden Programme genau das richtige für Sie. Das erste Programm zieht für Sie 6 Zahlen aus 49, während das zweite Programm eine Hilfe für den Mittwochstipper darstellt.

Die Programme gewähren dabei eine andere Zufälligkeit, als die der normalen Generatoren, da zunächst eine Zahl von Ziehungen festgelegt wird, und dann erst die Ziehung ausgewertet wird. Dabei kann man sich natürlich die Frage stellen, in wie weit dies für die Qualität der gezogenen Zahlen von Belang ist.

Sicher ist bis jetzt, daß man die Zahlen der nächsten Ziehung selbst mit einem Sharp PC nicht berechnen kann.

Variablenliste:

C:	Zahl aus 49/45
I:	Schleifenvariable
O:	Schleifenvariable
V:	Schleifenvariable
X:	Schleifenvariable
Y:	Zahl der Ziehungen
Z(I):	Zahlenspeicher

```
1 CLEAR
2 WAIT 300
3 PRINT"LOTTOZAHLEN"
4 PRINT"IM SPCS"
5 PRINT"6 AUS 49"
```

Do not sale !

```
6 DIM Z(7)
7 Y=RND 50
8 FOR I=1 TO Y
9 FOR X=1 TO 6
10 C=RND 49
11 IF C=0 THEN 10
12 FOR V=1 TO X-1
13 IF Z(V)=C THEN 10
14 NEXT V
15 Z(X)=C
16 NEXT X
17 NEXT I
18 FOR I=1 TO 5
19 FOR O=I+1 TO 5
20 IF Z(I)<Z(O) THEN 24
21 E=Z(O)
22 Z(O)=Z(I)
23 Z(I)=E
24 NEXT O
25 NEXT I
26 FOR I=1 TO 6
27 PRINT Z(I)
28 NEXT I
```

2.3.7 Lottozahlen 2

Das folgende Programm gehört zum Duo der Lottoprogramme in diesem Buch. Variablenliste und Erklärungen entnehmen Sie bitte den Seiten zuvor.

```
1 CLEAR
2 WAIT 300
3 PRINT"LOTTOZAHLEN"
4 PRINT"IM SPCS"
5 PRINT"6 aus 45"
6 DIM B(6)
7 Z=RND 50
8 FOR I=1 TO Z
```

Do not sale !


```
9 FOR W=1 TO 6
10 C=RND 45
11 IF C=0 THEN 10
12 FOR V=1 TO W-1
13 IF B(V)=C THEN 10
14 NEXT V
15 B(W)=C
16 NEXT W
17 NEXT I
18 FOR I=1 TO 5
19 FOR O=I+1 TO 5
20 IF B(I)<B(O) THEN 24
21 E=B(O)
22 B(O)=B(I)
23 B(I)=E
24 NEXT O
25 NEXT I
26 FOR I=1 TO 6
27 PRINT B(I)
28 NEXT I
```

Als Vorschlag zu diesen beiden Programmen möchte ich Ihnen Ziehungsprogramme für Fußballtoto und Rennquintett nennen. Die Durchführung dieser beiden Projekte stellt auch für den normalen Programmierer sicher kein Problem dar.

2.3.8 Zinseszins

Das unten aufgelistete Programm stellt eine klare Lösung zur Zinseszinsproblematik dar.

Der Rechner bietet Ihnen dabei den Vorteil, keine Nebenrechnungen durchführen zu müssen. Sie geben einfach die verlangten Werte an, und der Rechner gibt Ihnen sofort den Endbetrag. Somit ist dieses Programm weniger ein programmtechnisches Wunder, sondern nur eine Arbeitserleichterung.

Do not sale !

Variablenliste:

B:	Betrag
K:	Endbetrag
N:	Zeit in Jahren
Q:	Zwischenspeicher
Z:	Zinsperiode in Tagen

```
1 CLEAR
2 WAIT 150
3 PRINT"ZINSESZINS"
4 PRINT"IM SPCS"
5 INPUT"BETRAG";B
6 INPUT"JAHRESZINSFUSS:";P
7 INPUT"ZEIT ";N
8 PRINT"ZINSPERIODE IN"
9 INPUT"TAGEN:";Z
10 Z=360/Z
11 Q=1+P/100
12 K=B*Q^(N*Z)
13 K=(INT(K*100+0.5))/100
14 PRINT"DIE HOEHE DES "
15 PRINT"ENDBETRAGES IST"
16 PRINT K
17 INPUT"NEUE RECHNUNG:";A$
18 B$=LEFT$(A$,1)
19 IF B$="J" THEN 1
20 PRINT"ENDE"
21 END
```

Will man dieses Programm erweitern, so bieten sich die anderen Problematiken der Zinsrechnung natürlich besonders an. Die neuen Problemstellungen sollten so integriert werden, daß man in einem Menü unter den verschiedenen Möglichkeiten wählen kann.

2.3.9 Darlehen

Natürlich wünsche ich Ihnen nicht, in die Verlegenheit zu kommen, ein Darlehen aufnehmen zu müssen. Sollten Sie sich jedoch aus irgendeinem Grund entschließen, ein Darlehen aufzunehmen, dann sollten Sie vorher mit diesem Programm die richtige Höhe des Darlehens berechnen.

Beispiel:

Magnus will sich ein Auto kaufen. Er kann alle 2 Monate 355 DM aufbringen, und ist bereit dies 4 Jahre lang auf sich zu nehmen. Die Bank bietet ihm das Geld zu 17% Zinsen an. Wie hoch ist das Darlehen, das er aufnehmen kann?

Variablenliste:

W\$:	Zwischenspeicher
Z\$:	Antwort
D:	Höhe des Darlehens
E:	Zwischenspeicher
F:	Zwischenspeicher
I:	Jahreszinssatz
N:	Anzahl der Zahlungen pro Jahr
R:	Höhe der regelmäßigen Zahlungen
Y:	Laufzeit

```
1 CLEAR
2 WAIT 300
3 PRINT"DARALEHEN"
4 PRINT"IM SPCS"
5 PRINT"REGELM. ZAHLUNG:"
6 INPUT R
7 INPUT"LAUFZEIT: ";Y
8 PRINT"ZAHLUNGEN"
9 INPUT"PRO JAHR: ";N
10 INPUT"JAHRESZINSATZ: ";I
```

Do not sale !

```
11 I=I/100
12 E=(R*N)/I
13 F=((1+I/N)^(N*Y))
14 D=E*(1-(1/F))
15 PRINT"HOEHE DES "
16 PRINT"DARLEHENS="
17 PRINT D
18 INPUT"NEUE RECHNUNG ";Z$
19 W$=LEFT$(Z$,1)
20 IF W$="J" THEN 1
21 PRINT"ADIOS"
22 END
```

Dieses Programm ist eines der Programme, die man im Rahmen der Zins- und Zinseszinsproblematik zu einem Programmblock zusammenfassen kann. Wenn Sie sich für diese Thematik interessieren, dann sollten Sie sich das Programm Zinseszins in diesem Buch ansehen.

2.4 Programme zur Animation

7 Programme, die Spaß machen sollen und die Sie zum Schreiben von anderen interessanten Programmen anhalten sollen, sind in diesem Kapitel aufgelistet. Dabei stellen die Programme teilweise Rätsel und Aufgaben in Form von Programmen dar. Die Suche nach einem optimalen Lösungsweg wird dadurch interessanter, und Aufgaben in der Form eines Programmes machen mehr Spaß. Programmieren ist zwar eine ernste Angelegenheit, trotzdem soll der Spaß dabei nicht zu kurz kommen.

1. ReLi

ReLi ist die Abkürzung für "Von Rechts nach Links". Das Programm an sich ist recht einfach gestaltet. Man kann seinen Namen, oder auch ein anderes Wort, eingeben und der Computer druckt es von rechts nach links geschrieben aus.

2. Kegel

Kegel ist ein Rätsel in Programmform, von dem im oberen Text die Rede war. Man spielt gegen den Computer und es gibt nur einen Weg, immer zu gewinnen.

3. Spezroule

Wer kennt es nicht, das Russisch-Roulette? Man spielt mit dem Computer auf Leben und Tod. Eine spannende Simulation für alle, die etwas Phantasie haben.

4. PI

Die Berechnung der Zahl PI ist keine allzuschwere Angelegenheit. Man kann diese Pseudovariablen mit verschiedenen Methoden errechnen. Eine Methode zeigt dieses Programm. Die Aufgabe, die gleichzeitig gestellt wird, hat es allerdings in sich.

5. Orakel

In diesem Programm versucht sich Ihr Sharp PC als Wahrsager. Sie werden feststellen, daß er wirklich 100 prozentig richtige Weissagungen gibt.

6. Tage

In wieviel Tagen habe ich Geburtstag? Diese und andere Fragen kann das Programm "Tage" beantworten. Für alle, die auf ihren Geburtstag hinfiebern.

7. Temperaturen

Kelvin, Fahrenheit und Celsius mögen Ihnen als Maßeinheiten bekannt sein, aber wissen Sie auch in welchem Verhältnis diese zueinander stehen? Mit diesem Programm bekommen Sie einen Überblick über das Verhältnis dieser Einheiten. Haben Sie eigentlich schon einmal etwas von der Einheit Reaumur gehört?

2.4.1 Von Rechts nach Links

Dieses Programm gehört zu der Sorte von Programmen, die nur wenig Aufwand erfordern und dafür um so mehr Spaß bereiten. Wer hat es nicht schon einmal versucht, das Rückwärtssprechen? Wenn Sie zu den Kandidaten für das Weltbuch der Rekorde gehören, dann ist dieses Programm das absolute Muß in Ihrer Programmsammlung.

Überigens: Den Rekord im Rückwärtssprechen hält die Berlinerin Katja Nick. Sie spricht seit Ihrem zehnten Lebensjahr rückwärts. Nicht genug damit, sie dolmetscht auch noch fließend rückwärts. Also, wenn Sie nun ins Weltbuch der Rekorde wollen, dann sollten Sie dieses Programm sofort eintippen und üben, üben, üben...

Variablenliste:

A\$(1):	Name
B\$(1):	Name nach der Prozedur
N\$:	Antwort
I:	Schleifenvariable
X:	Länge des Strings

```
1 CLEAR
2 WAIT 150
3 PRINT"VON RECHTS..."
4 PRINT"NACH LINKS"
5 PRINT" IM SPCS "
```

Do not sale !

```
6 DIM A$(1)
7 DIM B$(1)
8 PRINT"NAME:"
9 INPUT A$(1)
10 X=LEN A$(1)
11 IF X>16 THEN PRINT"ZU LANG":GOTO 8
12 FOR I=X TO 1 STEP -1
13 B$(1)=B$(1)+MID$(A$(1),I,1)
14 NEXT I
15 WAIT 800
16 PRINT B$(1)
17 WAIT 150
18 INPUT"NEUE EINGABE: ";N$
19 IF LEFT$(N$,1)="J" THEN 1
20 END
```

2.4.2 Super Kegel-Spiel

Das vorliegende Spiel ist der Vorläufer des Bowlings. Der Ursprung dieses speziellen Spiels liegt in Dänemark. Das Programm ist aber keine Bowlingversion für die Sharp PC, sondern vielmehr ein Rätsel in der Form eines Programms.

Es sind 13 Kegel in einer Reihe nebeneinander aufgestellt. Sie haben nun immer 2 Möglichkeiten zu ziehen. Wenn Sie durch die Angabe einer ganzen Zahl einen Kegel bestimmen, dann wird genau dieser Kegel umgeworfen. Sollten Sie aber auf einen Zwischenraum zwischen zwei nebeneinanderstehenden Kegeln zielen, so werden beide Kegel umgeworfen.

Die Eingabe eines Wurfes, der auf Zwischenräume zielt, wird durch einen Kommawert gegeben (2.4: wirft 2 und 3 um). Eine Wurfeingabe wird nicht mit der ENTER-Taste beendet, sondern durch das Drücken der X-Taste. Ihre Aufgabe ist es, den letzten Kegel umzuwerfen. Dabei werfen abwechselnd Sie und der Computer.

Do not sale !

Variablenliste:

A\$: Kegelplätze
D\$: Zwischenspeicher
L\$: Antwort
M\$: Name des Spielers
C: Zwischenspeicher
G: Sieg oder Niederlage
I: Schleifenvariable
X: Zwischenspeicher

```
1 CLEAR:WAIT 150
2 DIM X(13)
3 DIM A$(13)
4 FOR I=1 TO 13:A$(I)="I":NEXT I
5 PRINT"SUPER-KEGEL"
6 PRINT"SPIEL"
7 PRINT" IM SPCS"
8 WAIT 50
9 INPUT"NAME: ";M$
10 FOR I=1 TO 12:PRINT A$(I);:NEXT I
11 PRINT A$(13)
12 D$="":C=0
13 B$=INKEY$:IF B$="" THEN 13
14 IF B$="X" THEN 17
15 D$=D$+B$
16 GOTO 13
17 C=VAL D$
18 IF C<1 OR C>13 THEN GOSUB 48:GOTO 10
19 IF C=INT C THEN LET A$(C)=" ":GOTO 22
20 A$(INT C)=" "
21 A$(INT C+1)=" "
22 FOR I=1 TO 13
23 IF A$(I)<>" " THEN 26
24 NEXT I
25 G=1:GOTO 51
26 GOTO 28
27 GOTO 10
28 PRINT"COMPUTERZUG"
```



```
29 N=0
30 FOR I=1 TO 12:PRINT A$(I);:NEXT I
31 PRINT A$(13)
32 WAIT 80
33 FOR I=1 TO 13
34 IF A$(I)="I" THEN LET N=N+1
35 IF A$(I)=" " THEN LET O=O+1
36 NEXT I
37 FOR I=X(1) TO 12
38 IF A$(I)="I" AND A$(I+1)="I" THEN LET A$(I)=" ":A$(I+1)=" ":GOTO 10
39 NEXT I
40 FOR I=13 TO 1 STEP -1
41 IF A$(I)="I" THEN LET A$(I)=" ":GOTO 47
42 NEXT I
43 FOR I=1 TO 13
44 IF A$(I)<>" " THEN 47
45 NEXT I
46 GOTO 51
47 GOTG 10
48 PRINT"FEHLER !!"
49 WAIT 150
50 RETURN
51 IF G=1 THEN PRINT "SIEGER - ";M$
52 IF G=0 THEN PRINT "SIEGER - SHARP"
53 WAIT 500
54 INPUT"NEUES SPIEL ";L$
55 IF L$="J" THEN 1
56 PRINT"BYE BYE..."
57 END
```

Da dieses Programm ein Rätsel ist, bietet es sich geradezu an, die optimale Lösung in die Computerroutine einzubauen. Das Problem liegt natürlich in der Tatsache, die optimale Lösung zu finden.

Do not sale !

2.4.3 Roulette Spezial

Nervenkitzel ersten Grades bietet dieses einfache, aber effektvolle Programm. Es simuliert das berühmte "Russische Roulette". Der Revolver hat 8 Schuß. Nachdem die Revolvertrommel gedreht worden ist, müssen abwechselnd Sie und der Computer abdrücken. Erscheint ein "KLICK" nachdem Sie abgedrückt haben, dann haben Sie für den Augenblick Glück gehabt. Mit ein wenig Phantasie bietet dieses Spiel spannende Unterhaltung.

Viel Glück!!

Variablenliste:

A\$:	Tastaturabfrage
J\$:	Antwort
X\$(8):	Kammerninhalte
F:	Kammernummer
I:	Schleifenvariable
V:	Kammer mit Kugel

```
1 CLEAR
2 WAIT 150:RANDOM
3 DIM X$(8)
4 PRINT"ROULETTE SPEZIAL"
5 WAIT150
6 PRINT"  IM  SPCS"
7 F=0
8 FOR I=1 TO 8
9 X$(I)="KLICK"
10 NEXT I
11 V=INT (RND*8)+1
12 X$(V)="BUMM!!"
13 F=F+1
14 PRINT"ICH SCHIESSE..."
15 PRINT X$(F)
16 IF X$(F)<>"KLICK" THEN 29
17 PRINT"DRUECK AB !"
```

Do not sale !

```
18 A$=INKEY$:IF A$="" THEN 18
19 F=F+1
20 PRINT"DU SCHIESST..."
21 PRINT X$(F)
22 IF X$(F)<>"KLICK" THEN 29
23 GOTO 13
24 PRINT"ICH BIN TOT!"
25 INPUT"NEUES SPIEL (J/N) ";J$
26 IF J$="J" THEN GOTO 1
27 END
28 PRINT"TUT MIR LEID!!"
29 PRINT"DU BIST TOT!!"
30 GOTO 25
```

2.4.4 Die Zahl PI

Dieses Programm hat einen sehr eindeutigen Hintergrund. Es berechnet die Zahl PI. Die Kreiszahl PI ist eine Pseudovariablen, die bei sehr vielen Berechnungen in der Physik und Mathematik benötigt wird. Dieses Programm berechnet die Zahl PI allerdings auch nicht genauer, als Sie bereits in Ihrem Sharp PC vorhanden ist. Insofern stellt dieses Programm mehr eine Aufgabe dar, als eine Problemlösung. Interessant wäre ein Programm, welches die Zahl PI auf n Stellen hinter dem Komma berechnen kann. Die Problematik liegt dabei in der Tatsache, daß der Sharp PC grundsätzlich nur auf 11 Stellen nach dem Komma ausgelegt ist. Sollte jemand ein Programm entwickelt haben, mit dessen Hilfe man n Stellen von PI berechnen kann, so wäre ich dankbar, wenn Sie mir über den Verlag ein Listing des Programms zukommen lassen würden.

Variablenliste:

A\$:	Tastaturabfrage
N\$:	Antwort
W\$:	Zwischenspeicher
G:	Zwischenspeicher
H:	Zwischenspeicher

Do not sale !

I: Schleifenvariable
K: Zwischenspeicher
L: Zwischenspeicher
M: Zwischenspeicher
P: Ergebnis

```
1 "D"
2 CLEAR
3 WAIT 150
4 PRINT"BERECHNUNG VON"
5 PRINT "    PI"
6 PRINT">> IM SPCS <<"
7 PRINT"S-> START"
8 WAIT 80
9 A$=INKEY$
10 IF A$="" THEN 9
11 IF A$<>"S" THEN 9
12 FOR I=1 TO 25 STEP 2
13 K=1/3^I
14 G=(1/I)
15 H=1/2^I
16 L=G*(H+K)
17 M=M+L
18 M=-M
19 P=ABS(M*4)
20 WAIT 150
21 PRINT P
22 NEXT I
23 INPUT"NOCHMAL (J/N):";N$
24 W$=LEFT$(N$,1)
25 IF W$="J" THEN 2
26 IF W$="N" THEN 29
27 PRINT"FALSCH EINGABE!"
28 GOTO 23
29 PRINT"BYE, BYE !!"
30 END
```

2.4.5 Das Orakel

Orakel besitzen allgemein den Ruf, alles zu wissen. In diesem Sinne ist dieses Programm auch ein kleines Orakel. Natürlich hat dieses kleine Orakel ein ganz besonderes Wissensgebiet.

Sie sollen nun zwei beliebige Zahlen aufschreiben, unter der Bedingung, nur die neun Ziffern und Null zu verwenden. Dabei darf man jede Ziffer nur einmal verwenden. Hier nur ein Beispiel:

123450
6789

Danach addieren Sie die die beiden Zahlen. Im genannten Beispiel ergibt das:

130239

Nun dürfen Sie eine beliebige Ziffer aus dem Ergebnis streichen. Nehmen wir einmal an, daß wir die Ziffer 2 aus unserem Ergebnis streichen.

Nun kommt das Orakel zum Zuge. Als erstes möchte es wissen, wieviel Stellen das Ergebnis ohne die gestrichene Zahl hat. In unserem Beispiel sind es 5 Stellen. Nun müssen wir dem Orakel noch die 5 Ziffern angeben (Also: 1,3,0,3,9).

Das Orakel ist nun in der Lage, Ihnen die Zahl zu nennen, die Sie gestrichen haben. Mit diesem Programm werden Sie sicher auch den letzten Zweifler von der Leistungsfähigkeit Ihres Sharp PC überzeugen.

Eine Anmerkung zur Variablenliste dieses Programms: Die Variablenliste ist in diesem Fall sehr allgemein gehalten, denn Sie sollen die Arbeitsweise des Orakels nicht so leicht durchschauen.

Do not sale !

Varablenliste:

L\$:	Name des Spielers
N\$:	Antwort
W\$:	Zwischenspeicher
A:	Zwischenspeicher
B:	Zwischenspeicher
E:	Zwischenspeicher
I:	Schleifenvariable
Q:	Zwischenspeicher
R:	Zwischenspeicher
S:	Anzahl der Stellen

```
1 CLEAR
2 WAIT 150
3 PRINT"DAS ORAKEL"
4 PRINT"IM SPCS"
5 INPUT"WIE HEISST DU: ";L$
6 IF LEN L$>6 THEN PRINT"ZU LANG": GOTO 5
7 PRINT"WIEVIEL STELLEN"
8 INPUT"HAT DIE SUMME: ";S
9 IF S=10 THEN 32
10 IF S>10 OR S<5 THEN 35
11 FOR I=1 TO S
12 PRINT I;" ZAHL: ";:INPUT A
13 B=B+A
14 NEXT I
15 FOR Q=1 TO 10
16 R=Q*9
17 IF R=Q*9 THEN LET E=0: GOTO 22
18 IF R>B THEN 20
19 NEXT Q
20 E=Q*9-B
21 B=B*SIN(Q*E)
22 PRINT"ACHTUNG !!!"
23 PRINT L$
24 PRINT"DAS ORAKEL SAGT"
25 PRINT"DIE GESUCHTE "
26 PRINT"ZAHL IST=";E
```

```
27 INPUT"NEUES ORAKEL: ";N$
28 W$=LEFT$(N$,1)
29 IF W$="J" THEN 1
30 PRINT"GOODBYE !!!"
31 END
32 PRINT L$;" DU SOLLST
33 PRINT"2 ZAHLEN BILDEN"
34 GOTO 8
35 PRINT L$
36 PRINT"BIITE LIES DEN"
37 PRINT"TEXT NOCHEINMAL"
38 PRINT"GENAU !!!"
39 PRINT"SEE YOU LATER.."
40 END
```

Der Vorschlag zu diesem etwas ungewöhnlichen Programm bezieht sich auf die Lösung der Rätselaufgabe, die in diesem Programm versteckt ist. Wenn Sie das Geheimnis gelöst haben, dann werden Sie auch ohne Programm in der Lage sein, Orakel zu spielen.

2.4.6 Tage

Gehören Sie auch zu den Menschen, die das ganze Jahr über auf Ihren Geburtstag hinfiebern? Wenn ja, dann ist dieses Programm der Schlüssel zur Gewißheit. Mit Hilfe dieses Programms können Sie genau die Tage, Stunden, Minuten und Sekunden bis zu Ihrem Geburtstag ermitteln.

Wenn Sie zum Beispiel am 27. April Geburtstag haben, dann geben Sie einfach 27.04 ein und danach in gleicher Form das heutige Datum. Der Rechner gibt dann nach kurzer Rechendauer die Zahl der Tage u.s.w. an.

Variablenliste:

B:	Geburtstag
C:	Datum
D:	Zwischenspeicher
E:	Zwischenspeicher
F:	Zwischenspeicher
G:	Zwischenspeicher
H:	Zwischenspeicher
I:	Schleifenvariable
K:	Zwischenspeicher
L:	Zwischenspeicher
M:	Zwischenspeicher
N:	Zwischenspeicher
P:	Zwischenspeicher
Q:	Zwischenspeicher
T:	Ergebnis in Tagen

```
1 CLEAR
2 WAIT 150
3 PRINT"TAGE BERECHNUNG"
4 INPUT"GEBURTSTAG:";B
5 INPUT"HEUTIGER TAG:";C
6 G=((C*100)-INT(C*100))*100
7 G=INT(G+0.5)
8 E=C-INT C
9 E=(INT(E*100+0))/100
10 F=((B*100)-INT(B*100))*100
11 F=INT(F+0.5)
12 D=B-INT B
13 D=(INT(D*100+0))/100
14 H=G-F
15 K=D*100
16 IF INT C <INT B THEN 18
17 GOTO 22
18 IF B-D>30 THEN GOSUB 26
19 C=C+30:C=C-0.01:E=(C-INT C)
20 E=(INT(E*100+0))/100
21 GOTO 27
```



```
22 IF C-E>30 THEN GOSUB 25
23 IF B-D>30 THEN GOSUB 26
24 GOTO 27
25 C=30:RETURN
26 B=30:RETURN
27 IF D>E THEN LET E=E+0.12:H=H-1
28 Q=E-D:Q=INT(Q*100+0.5)
29 P=INT C-INT B
30 IF D<>E THEN 33
31 GOTO 40
32 REM MONAT MIT 31 TAGEN
33 FOR I=1 TO Q:K=INT(K+0.5)
34 IF K=2 THEN LET L=L-2
35 FOR J=1 TO 7:READ A
36 IF K=A THEN LET L=L+1
37 NEXT J:RESTORE:K=K+1
38 IF K>12 THEN LET K=1
39 NEXT I
40 IF F<>G THEN 43
41 GOTO 46
42 REM SCHALTJAHR
43 FOR I=F+1 TO F+H
44 IF (I/4)-INT(I/4)=0 THEN LET M=M+1
45 NEXT I
46 T=Q*30+P+(H*365)+L+M
47 REM AUSGABE
48 PRINT"ALTER=";T;" TAGE"
49 N=T*24:PRINT"ALTER=";N;" STD."
50 N=N*60:PRINT"ALTER=";N;" MIN."
51 N=N*60:PRINT"ALTER=";N;" SEC."
52 END
53 DATA 1,3,5,7,8,10,12
```

Dieses Programm kann natürlich als Grundlage für andere genaue Datumsberechnungen benutzt werden. Eine naheliegende Anwendung wäre die Berechnung Ihres Alters in Tagen.

Do not sale !

2.4.7 Temperaturen

Eine Problematik, die der der yards-Umrechnung nicht unähnlich ist, ist die Umrechnung der verschiedenen Temperatursysteme.

In der Physik findet die Kelvinskala Anwendung. Von daher ist sie auch als Grundlage für die beiden anderen Temperatursysteme zu sehen. Auf dem europäischen Festland ist die Celsiusskala gebräuchlich und in England und Amerika gibt es Fahrenheit. Eine vierte sehr wenig bekannte Temperaturmaßeinheit ist das Reaumur.

Begründer dieser Einheit war der französische Physiker Rene-Antoine Ferchault de Reaumur (* 28.2.1683/ + 17.10.1757). Diese Skala reicht von 0 Grad bis 80 Grad Reaumur, wobei 0 Reaumurgrad den 0 Grad Celsius entspricht und die 80 Reaumurgrad den 100 Grad Celsius. Die Beziehungen der einzelnen Skalen untereinander können Sie der folgenden Tabelle entnehmen.

absoluter Nullpunkt	Eis- punkt	Siedepunkt des Wassers	
0	273,15	373,15	Kelvin
273,16	0	100	Celsius
	31	212	Fahrenheit
	0	80	Reaumur

Variablenliste:

A\$:	Tastaturabfrage
G\$:	Platzhalter für Fahrenheit
N\$:	Antwort
W\$:	Zwischenspeicher
C:	Grad Celsius
F:	Grad Fahrenheit
K:	Grad Kelvin

Do not sale !

```
1 CLEAR
2 WAIT 150
3 PRINT"TEMPERATUREN"
4 PRINT">> IM SPCS <<"
5 C=20
6 K=C+273.15
7 F=31+C*1.81
8 G$=STR$(F)
9 IF F<31 OR F>212 THEN LET F=999
10 IF F=999 THEN LET G$="-"
11 PRINT C;"C/ ";K;"K/ "G$;"F"
12 A$=INKEY$
13 IF A$="" THEN 12
14 IF A$="Q" THEN LET C=C+0.05
15 IF A$="W" THEN LET C=C+1
16 IF A$="E" THEN LET C=C+10
17 IF A$="R" THEN LET C=C+100
18 IF A$="Z" THEN LET C=C-0.05
19 IF A$="X" THEN LET C=C-1
20 IF A$="C" THEN LET C=C-10
21 IF A$="V" THEN LET C=C-100
22 IF A$="P" THEN 25
23 IF C<-273.15 THEN LET K=0:C=-273.15:GOTO 11
24 GOTO 6
25 INPUT"SICHER ";N$
26 W$=LEFT$(N$,1)
27 IF W$="J" THEN 29
28 IF W$="N" THEN 11
29 PRINT"ADIOS !!"
30 END
```

Der Vorschlag zu diesem Programm bezieht sich auf die Integrierung der Reaumur-Skala in dieses Programm. Die Anpassung der Reaumur-Skala erfolgt nach demselben System wie die der Fahrenheitskala.

Do not sale !

2.5 Programme für das Hobby

Das Hobby, das in diesem Kapitel hauptsächlich angesprochen wird, ist das Hobby Computer. Hilfsroutinen für das Computern, aber auch ganz anders geartete Programme geben sich hier ein Stelldichein. Natürlich wäre es ein leichtes gewesen, 20 oder 30 Programme zu erstellen, die das Computern erleichtern und einem lästige Arbeiten abnehmen. Aber obwohl dieses Buch ein Buch für Computer ist, ist dieser Themenbereich nur einer von insgesamt 6.

1. Monitor

Mit einem Hilfsprogramm fängt dieses Kapitel an. Mit diesem Programm kann man Programme in Maschinensprache erstellen. Die Zahlen werden in dezimaler Form dargestellt. Ein Programm zur Umsetzung in andere Zahlensysteme ist das Programm Systeme.

2. Weltweit

Entfernungen von einem Punkt auf dem Erdball zu einem beliebigen anderen Punkt berechnet das Programm Weltweit. Nur einen Atlas und dieses Programm werden benötigt und schon geht es los.

3. RND-RND

Jeder, der sich etwas mit seinem Sharp beschäftigt hat, weiß, daß mit der RND-Funktion Zufallszahlen erzeugt werden. Daß man solche Zahlen auch ohne diesen Befehl erzeugen kann und wie man so etwas macht, zeigt dieses Programm.

4. RND Test

Wie zufällig sind die Zufallszahlen aus meinem Sharp eigentlich? Dieser Frage geht dieses Programm auf den Grund, welches die Verteilung der Zufallszahlen überprüft.

5. Shuttle

Haben Sie sich schon einmal gefragt, wie schnell ein Satellit in seiner Umlaufbahn fliegt? Diese Frage beantwortet das Programm Shuttle. Außerdem gibt es noch die Zeit an, die der Satellit für eine Erdumrundung benötigt.

6. Systeme

Das man Zahlen in verschiedenen Zahlensystemen darstellen kann und auch so darstellt, dürfte jedem Computerfreak bekannt sein. Umrechnungen vom Dezimalsystem in jedes beliebige System bis zum Hexadezimalsystem erledigt dieses Programm.

7. Bubble

Das Sortieren von Daten ist eine wichtige Funktion bei Dateiverwaltungsprogrammen. Wie man Daten sortiert, zeigt das Programm Bubble. Das Programm arbeitet dabei nach dem Bubblesort-System.

8. Primzahlen

Ob eine Zahl eine Primzahl ist oder nicht, das errechnet dieses Programm. Es generiert in einem beliebigen Zahlenbereich die dort vorkommenden Primzahlen.

2.5.1 Monitor

Monitor ist eine kleines, aber sehr leistungsfähiges Monitorprogramm im SPCS. Mit Hilfe dieses Programms kann man Programme in Maschinensprache erstellen und untersuchen. Der Monitor arbeitet mit Dezimalzahlen. Das Programm hat folgende Funktionen:

1. L: Laden von Maschinensprachprogrammen
2. S: Speichern von Speicherinhalten
3. X: Abbrechen der Speicherbearbeitung
4. Q: Beenden des Programms

Nach dem Programmstart verlangt das Programm die Eingabe der Anfangsadresse und der Endadresse des zu bearbeitenden Speicherbereichs. Dann wird im Display die Adresse und deren Speicherinhalt gezeigt. Man hat nun 3 Möglichkeiten, weiter zu verfahren:

1. Man drückt ENTER - Der Rechner verändert den Speicher nicht und zeigt die nächste Adresse samt Speicherinhalt an.
2. Man gibt einen Wert zwischen 0 und 255 ein - der Rechner verändert den Speicherinhalt auf den angegebenen Wert, und zeigt dann die nächste Adresse an.
3. Man gibt einen der 4 Systembefehle an, der dann wahrscheinlich auch entsprechend ausgeführt wird.

Variablenliste:

M\$:	Zwischenspeicher
NS:	Eingabe im Arbeitsmodus
Q\$:	Antwort
T\$:	Antwort
B:	Inhalt der Adresse
E:	Endadresse

U: Startadresse beim Speichern
V: Endadresse beim Speichern
S: Startadresse
Z: Adresse

```
1 CLEAR
2 WAIT 150
3 PRINT"DEZ-MONITOR"
4 PRINT"IM SPCS"
5 INPUT"START:",S
6 INPUT"ENDE:",E
7 FOR Z=S TO E
8 B=PEEK Z
9 PRINT"ADR: ";Z;" ";B;
10 INPUT N$
11 IF N$="" THEN 18
12 IF N$="X" THEN 5
13 IF N$="L" THEN 31
14 IF N$="S" THEN 20
15 IF N$="Q" THEN 37
16 N=VAL N$
18 NEXT Z
19 GOTO 1
20 WAIT 300
21 PRINT"SPEICHERN"
22 INPUT"START - ";U
23 INPUT"ENDE - ";V
24 INPUT"FILENAME ";Q$
25 INPUT"ALLES BEREIT ";T$
26 IF T$<>"J" THEN 1
27 SAVEM Q$,U,V
28 PRINT Q$;"...":WAIT300
29 PRINT"GESPEICHERT "
30 GOTO 3
31 WAIT 300
32 PRINT"LADEN..."
33 INPUT"FILENAME: ";Q$
34 LOADM Q$
35 PRINT Q$;" GELADEN ":'
```

Do not sale !

```
36 GOTO 3
37 PRINT"BIST DU SICHER ?"
38 INPUT M$
39 W$=LEFT$(M$,1)
40 IF W$<>"J" THEN 1
41 PRINT"ARRIVEDERCI"
42 END
```

Bekanntlich ist es einfacher, bei der Maschinensprachprogrammierung mit Hexadezimalzahlen zu arbeiten. Daher mein Vorschlag:

Ändern Sie das Programm so ab, daß alle Werte in Hexadezimalzahlen ausgegeben werden. Hilfe zur Lösung finden Sie bei meinem Programm SYSTEME in diesem Buch.

2.5.2 Weltweit

Dieses Programm ist ein gefundenes Fressen für alle, die ihren Finger gerne mal über die Landkarte streichen lassen. Auch reisefreudige Zeitgenossen werden die Vorzüge dieses Programms sehr schnell zu schätzen wissen.

Mit Hilfe dieses Programms sind Sie in der Lage, die Entfernung zwischen zwei beliebigen Orten auf dieser Erde genau zu berechnen.

Sie müssen nur den Atlas zur Hand nehmen und die Längen- und Breitengrade der Orte möglichst genau angeben. Der Rechner gibt dann die genaue Entfernung dieser beiden Orte an.

Variablenliste:

X\$:	Name erster Ort
Y\$:	Name zweiter Ort
Z\$:	Antwort
A:	Länge in Grad

Do not sale !

B: Länge in Minuten
C: Breite in Grad
D: Breite in Minuten
E: Entfernung
H: Zwischenspeicher
L: Zwischenspeicher
M: Zwischenspeicher
N: Zwischenspeicher
W: Konstante zur Umrechnung von Bogemaß

```
1 CLEAR
2 WAIT 150
3 W=0.174533
4 PRINT"WELTWEIT"
5 PRINT"EIN PROGRAMM"
6 PRINT"IM SPCS"
7 INPUT"NAME ORT 1:";X$
8 INPUT"LAENGE 1(GRAD):";A
9 INPUT"LAENGE 1(MIN.):";B
10 INPUT"BREITE 1(GRAD):";C
11 INPUT"BREITE 1(MIN.):";D
12 L=A+B/60
13 H=C+D/60
14 INPUT"NAME ORT 2:";Y$
15 INPUT"LAENGE 2(GRAD):";A
16 INPUT"LAENGE 2(MIN.):";B
17 INPUT"BREITE 2(GRAD):";C
18 INPUT"BREITE 2(MIN.):";D
19 M=A+B/60
20 N=C+D/60
21 F=COS(W*L)
22 E=111.2*SQR((L-M)^2+(H-N)^2*F*COS(W*M))
23 PRINT"DIE ENTFERNUNG"
24 PRINT"ZWISCHEN"
25 PRINT X$
26 PRINT"UND"
27 PRINT Y$
28 PRINT"BETRAEGT:"
29 PRINT E;" KM"
```

Do not sale !

```
30 INPUT"NEUE BERECHNUNG:";Z$
31 IF LEFT$(Z$,1)="J" THEN 1
32 PRINT"CIAO"
33 END
```

Wenn Sie das Programm des öfteren benutzen, dann sollten Sie die Koordinaten Ihres Heimatortes fest im Programm implementieren.

2.5.3 RND-RND

In allen Rechnern der Sharp PC Reihe ist der RND-Befehl implementiert. Mit Hilfe dieses Befehls kann man Pseudozufallszahlen erzeugen. Das Programm "RND-RND" erzeugt ebenfalls Zufallszahlen. Dabei arbeitet das Programm nach dem gleichen Prinzip wie die RND-Funktion. Nach dem Start verlangt das Programm eine Grundlage nach der die Zufallszahlen berechnet werden. Allerdings führt dieses Programm weiter. Nachdem es 100 Zufallszahlen generiert hat, gibt es die Verteilung der einzelnen Zufallswerte auf die Werte 0 bis 9 aus. Danach gibt es die durchschnittliche Verteilung der Zufallszahlen aus. Durch diese beiden zusätzlichen Funktionen hat man die Möglichkeit, die Ergebnisse des Generators zu kontrollieren. Jeder der sich für diese Thematik interessiert, sollte das Programm RND-Test ausprobieren. Mit Hilfe dieses Programms kann man die Zuverlässigkeit des eingebauten Zufallszahlengenerators überprüfen.

Variablenliste:

F(X):	Verteilerspeicher
G:	Grundlage
H:	Zwischenspeicher
Q:	Durchschnitt
T:	Tempo der Bildschirmausgabe
X:	Zählvariable
Z:	Zählvariable

Do not sale !

```
1 CLEAR
2 WAIT 150
3 PRINT"ZUFALLSZAHLN"
4 PRINT"OHNE RND"
5 PRINT"IM SPCS"
6 INPUT"GRUNDLAGE:";G
7 DIM F(9)
8 FOR X=0 TO 9
9 F(X)=0
10 NEXT X
11 INPUT"TEMPO (1-10):";T
12 IF T<1 OR T>10 THEN 11
13 T=T*20
14 WAIT T
15 Z=0
16 H=997*G
17 H=H-INT H
18 Z=Z+1
19 PRINT INT 10*H
20 X=INT 10*H
21 F(X)=F(X)+1
22 G=H
23 IF Z<100 THEN 16
24 FOR X=0 TO 9
25 PRINT "F(";X;")=";F(X)
26 NEXT X
27 INPUT"F(X) NOCHMAL:";M$
28 IF LEFT$(M$,1)="J" THEN 24
29 Q=0
30 FOR X=0 TO 9
31 Q=Q+(F(X)-100)*(F(X)-100)
32 NEXT X
33 Q=Q/1000
34 PRINT"VERTEILUNG IM"
35 PRINT"DURCHSCHNITT=";Q
36 WAIT 150
```

Do not sale !

```
37 INPUT"NEUE RECHNUNG ";M$
38 IF LEFT$(M$,1)="J" THEN 1
39 PRINT"ENDE"
40 END
```

2.5.4 RND Test

Die Zahlen, die mit dem RND-Befehl erzeugt werden, sind bekanntlich keine echten Zufallszahlen, sondern Pseudozufallszahlen. Dabei wird versucht, eine optimale Verteilung der Zufallswerte zu erreichen. Wenn also Zufallszahlen von 0 bis 10 generiert werden sollen, dann soll die Häufigkeit der einzelnen Werte gleich sein.

Das Programm RND Test soll, wie der Name schon sagt, überprüfen, in wie weit das Ziel der optimalen Verteilung erreicht wird. Das Programm läßt 1000 Zufallszahlen erzeugen und speichert die Häufigkeit des Vorkommens der einzelnen Zahlen. Man hat die Auswahl, ob man die Ziehung am Display mitverfolgen will, oder ob man auf das Endergebnis warten will.

Überigens: Wer sich die Ziehung am Display ansehen will, sollte Zeit und Geduld mitbringen, denn die Ziehung nimmt einige Minuten in Anspruch.

Variablenliste:

A\$:	Ausgabe Ja, Nein
N\$:	Neue Rechnung
W\$:	Zwischenspeicher
A:	RND Zahl
A(I):	Speicher der einzelnen Zahlen

```
1 CLEAR
2 WAIT 300
3 DIM A(10)
4 PRINT"RND TEST"
```

```
5 PRINT"IM SPCS"
6 INPUT"AUSGABE:";A$
7 W$=LEFT$(A$,1)
8 IF W$="J" THEN B=1:GOTO 19
9 IF W$="N" THEN B=0:GOTO 12
10 PRINT"FEHLER !!!"
11 GOTO 6
12 PRINT"BITTE WARTEN"
13 FOR I=1 TO 1000
14 A=RND 10
15 A=INT A
16 A(A)=A(A)+1
17 NEXT I
18 GOTO 26
19 PRINT"ACHTUNG !":WAIT 50
20 FOR I=1 TO 1000
21 A=RND 10
22 A=INT A
23 A(A)=A(A)+1
24 PRINT A
25 NEXT I
26 PRINT"ERGEBNISSE:"
27 FOR I=0 TO 9
28 PRINT "A(";I;")=";A(I)
29 NEXT I
30 INPUT"NEUE RECHNUNG ";N$
31 W$=LEFT$(N$,1)
32 IF W$="J" THEN 1
33 PRINT"CIAO"
34 END
```

2.5.5 Space Shuttle

Das Space Shuttle der NASA ist nur eines von vielen Objekten, die sich ständig oder gelegentlich im Orbit unseres Planeten befinden. Für alle, die sich für Satelliten oder ähnliches interessieren, dürfte dieses Programm besonders interessant sein. Es berechnet die Geschwindigkeit und die Umlaufzeit von Objekten, die sich in der Umlaufbahn unserer Erde befinden.

Do not sale !

Nachdem man das Programm gestartet hat, muß man die Flughöhe und die Masse des Objektes angeben. Natürlich müssen die Angaben nicht hundertprozentig stimmen, aber je genauer die Angaben, desto genauer ist das Ergebnis der Berechnung. Sollten Sie keine Informationen über die Masse und die Flughöhe von Satelliten besitzen, dann wenden Sie sich bitte an das Bundesministerium für Raumfahrt. Sollte Ihnen dieser Weg zu umständlich sein, so werden Sie in Ihrer Stadtbücherei sicherlich auch an die notwendigen Daten kommen.

Variablenliste:

J	Zwischenspeicher
G	Durchschnittliche Gravitation
H	Flughöhe des Objektes
M	Masse des Objektes
Q	Zwischenspeicher
R	Konstante
T	Zwischenspeicher
V	Geschwindigkeit
W	Zwischenspeicher
Y	Zwischenspeicher
Z	Umlaufzeit in Minuten

```
1 CLEAR
2 WAIT 300
3 R=6378388
4 G=9.81
5 PRINT"SPACE SHUTTLE"
6 PRINT"IM SPCS"
7 INPUT"FLUGHOEHE:";H
8 INPUT"MASSE:";M
9 W=G/(H*1000+R)
10 V=R*SQR W
11 T=R+H*100
12 Q=M*V^2/T
13 J=H*1000+R
```

```
14 Y=2*PI*J
15 Z=Y/V
16 Z=Z/60
17 PRINT"GESCHWINDIGKEIT:"
18 PRINT V;"M/S"
19 PRINT"UMLAUFZEIT:"
20 PRINT Z;"MIN."
21 INPUT"NEUE RECHNUNG:";M$
22 E$=LEFT$(M$,1)
23 IF E$="1" THEN 1
24 PRINT"BYE BYE"
25 END
```

Vorschlag:

Wenn Sie zu den Leuten gehören, die Programme gerne verbessern oder erweitern, dann möchte ich Ihnen folgenden Vorschlag machen: Lassen Sie die Geschwindigkeit nicht mehr in M/S, sondern in KM/H berechnen. Die Umlaufzeit könnte man in Stunden und Minuten angeben.

2.5.6 Systeme

Mit Zahlen zu arbeiten, ist für jeden modernen Menschen eine Selbstverständlichkeit, aber den wenigsten ist dabei bewußt, daß es sich bei den Zahlen, mit denen sie umgehen, um Zahlen des Dezimalsystems handelt. Die Zahl 10 in unserem Dezimalsystem (10er-System) ist uns zwar sehr geläufig und daher auch praktisch, das ändert aber nichts an der Tatsache, daß die Zahl 10 willkürlich gewählt wurde. Jeder Computerfreak weiß, daß bei der Datenverarbeitung das Binärsystem (2er-System) und das 16er-System (Hexa- oder Sedezimalsystem) Anwendung finden.

In der Zeitmessung und bei den englischen Maßen finden wir das Duodezimalsystem (12er-System) und beim Führen von Gefängniskalendern Ansätze des 5er-Systems.

Do not sale !

Das folgende Programm wandelt Zahlen aus dem Zehnersystem in ein beliebiges Zahlensystem von 2 - 16 um. Da das Ergebnis als String aufgebaut ist, ist das Rechnen mit diesen Zahlen in diesem Programm nicht möglich.

Nachdem das Programm gestartet worden ist, verlangt der Rechner nach der Ziffernzahl, also nach dem Zahlensystem, das wir berechnen wollen. Dann fordert er uns auf, die Zahl einzugeben, die umgewandelt werden soll. Nachdem diese beiden Werte eingegeben worden sind, gibt der Rechner das entsprechende Ergebnis aus.

2.5.7 Bubble

Das Programm Bubble ist ein Programm, mit dessen Hilfe man bis zu 30 Namen alphabetisch ordnen kann. Die Anzahl der Namen kann man zu Beginn des Programms selbst bestimmen. Diese Anwendung ist zwar auch so sehr interessant, aber eigentlich soll dieses Programm zeigen, wie man in BASIC sortieren kann. Der Kernteil des Programms kann sehr leicht in andere Programme eingebunden werden und ist dadurch eine Hilfe bei eigenen Programmen. Die eigentliche Routine liegt zwischen Zeile 12 und Zeile 19 einschließlich. Besonders in Programmen, die Daten verwalten, sind Sortiervorgänge von großer Bedeutung. Die Namen werden in dem in Zeile 8 dimensionierten String M\$ abgelegt und werden dann der Reihe nach geordnet. Nachdem die Namen geordnet worden sind, werden sie nacheinander in alphabetischer Reihenfolge auf dem Display ausgegeben. Wer sich die Ergebnisse nochmal ansehen will kann dies beliebig oft tun. Für diesen Fall fragt das Programm nach jeder Ausgabe der Ergebnisse, ob man sich diese nocheinmal ansehen möchte.

Variablenliste:

E\$:	Zwischenspeicher
M\$():	Namen
N\$:	Antwort

Do not sale !

W\$: Zwischenspeicher
L: Schleifenvariable
N: Anzahl der Namen
T: Schalter

```
1 CLEAR
2 WAIT 150
3 PRINT">> BUBBLE <<"
4 PRINT">> IM SPCS <<"
5 PRINT"ANZAHL DER"
6 INPUT"NAMEN:";N
7 IF N<1 OR N>30 THEN 5
8 DIM M$(N+1)
9 FOR L=1 TO N
10 INPUT"NAME:";M$(L)
11 NEXT L
12 T=0
13 FOR L=1 TO N
14 IF M$(L)<=M$(L+1) THEN 19
15 E$=M$(L)
16 M$(L)=M$(L+1)
17 M$(L+1)=E$
18 T=1
19 NEXT L
20 WAIT 60
21 IF T=1 THEN 12
22 FOR L=1 TO N+1
23 PRINT M$(L)
24 NEXT L
25 WAIT 150
26 INPUT"ERG.NOCHMAL:";N$
27 W$=LEFT$(N$,1)
28 IF W$="J" THEN 22
29 IF W$="N" THEN 32
30 PRINT"FALSCH EINGABE"
31 GOTO 26
32 INPUT"NEUE RECHNUNG:";N$
33 W$=LEFT$(N$,1)
```

Do not sale !

```
34 IF WS="J" THEN 1
35 IF WS="N" THEN 38
36 PRINT"FALSCHE EINGABE"
37 GOTO 32
38 PRINT"TSCHUESS!!"
39 END
```

Dies ist natürlich nur eine von vielen möglichen Sortiermöglichkeiten. Wenn Sie etwas Zeit haben, dann könnten Sie sich noch ein Paar Gedanken über Sortierverfahren machen. Der Unterschied zwischen verschiedenen Verfahren liegt ja nicht nur in der Verfahrensweise an sich, sondern im Wesentlichen in der Geschwindigkeit. Während das eine Programm für eine Aufgabe 4 Stunden benötigt, löst ein anderes Programm auf dem gleichen Rechner die Aufgabe in wenigen Minuten. Sie wissen ja: Zeit ist Geld!

2.5.8 Primzahlen

Gibt es unendlich viele Primzahlzwillinge? Diese Frage beschäftigt noch, oder besser gerade heute, Mathematiker, die mit Hochleistungscomputern, wie dem CRAY immer größere Primzahlen berechnen, um neue Anhaltspunkte in dieser Problematik zu erhalten. Für diejenigen unter Ihnen, die mit dieser Problematik nicht so vertraut sind, möchte ich eine kurze Definition des Problems geben.

Primzahlen sind natürliche Zahlen, die durch keine ganze Zahl, außer durch 1 oder sich selbst, teilbar sind. Es gibt unendlich viele Primzahlen und ihre gesetzmäßige Aufeinanderfolge ist nicht bekannt. Die Verteilung der Primzahlen unter den natürlichen Zahlen ist sehr unregelmäßig.

Ein sehr häufig auftretender Fall sind Primzahlzwillinge, also 2 im Abstand von 2 auftretende Primzahlen (2 u. 3, 5 u. 7). Ungeklärt ist bis heute, ob es auch unendlich viele Primzahlzwillinge gibt.

Die Antwort auf diese Frage werden wir mit unseren Rechnern wohl kaum beantworten. Nichtsdestotrotz habe ich ein Programm im SPCS entworfen, mit dessen Hilfe man Primzahlen generieren kann. Der Rechner fragt zunächst nach dem Intervallanfang, in dem er Primzahlen suchen soll. Danach verlangt er nach der Größe des Intervalls.

Nachdem die Werte eingegeben worden sind, erscheint die geordnete Abfolge der Primzahlen auf Ihrem Display. Somit sind Sie in der Lage, ausgewählte Zahlenwerte auf ihre Eigenschaft als Primzahl hin zu untersuchen.

Variablenliste:

M\$:	Zwischenspeicher
N\$:	Antwort
A:	Intervallbegin
B:	Intervallende
E:	Differenz
I:	Schleifenvariable
Q:	Schleifenvariable
X:	Zwischenspeicher

```
1 CLEAR
2 WAIT 150
3 PRINT"PRIMZAHLEN"
4 PRINT"IM SPCS"
5 INPUT"START:";A
6 INPUT"ENDE:";B
7 FOR I=A TO B
8 FOR Q=2 TO I-1
9 X=I/Q
10 E=A-INT A
11 IF E=0 THEN NEXT I
12 NEXT Q
13 PRINT I
14 Z$=INKEY$
```

Do not sale !

```
15 IF Z$="X" THEN 17
16 NEXT I
17 PRINT"NEUE BERECHNUG:";N$
18 M$=LEFT$(N$,1)
19 IF M$="J" THEN 1
20 PRINT"ENDE..."
21 END
```

Vorschlag:

Dieses Programm bietet zwar eine komfortable Berechnung von Primzahlen, aber die eigentliche Problematik der Primzahlzwillinge ist nicht optimal gelöst.

Erweitern Sie das Programm so, daß nur noch Primzahlzwillinge angegeben werden.

2.6 Programme zum Spielen

Obwohl Computer ja eigentlich als Rechenmaschinen konzipiert worden sind, wird auf jedem Computersystem, auf dem es irgendwie möglich ist, gespielt. Dieser Spieltrieb des Anwenders ist selbst noch bei Leuten zu finden, die Tag für Tag mit diesen Geräten ernsthaft arbeiten. Dabei ist noch eine andere interessante Besonderheit zu finden. Obwohl es inzwischen für viele Homecomputer perfekte Videospiele gibt, werden noch immer Spiele wie das gute alte Hangman gespielt. Die Spiele, die in diesem Kapitel zusammengestellt worden sind, sind zwangsläufig ohne Graphik und Sound. Trotzdem macht es unheimlich Spaß, im Zug oder in der Schulstunde eine Runde Hangman zu spielen, oder eine Zahlenkombination zu knacken.

1. Henker

Henker ist eine klassische Hangmanversion mit modernen Begriffen aus der Computerbranche. Obwohl man die Begriffe

beim Eintippen sieht, hat man große Schwierigkeiten die Begriffe nacher zu erraten.

2. Jackpot

Wie mitten in Las Vegas fühlt man sich, wenn man am Einarmigen Banditen spielt. Das Programm simuliert einen Einarmigen Banditen mit drei rotierenden Walzen.

3. Schloß

Ein Zahlenschloß zu knacken ist gar nicht so einfach. Aber der Computer gibt Hilfestellungen. Wer kann die Kombination am schnellsten knacken.

4. Rekneh

Spezielle Hangmanversion, bei der die zu erratenden Begriffe von rechts nach links dargestellt sind. Wem Henker zu einfach ist, der sollte Rekneh spielen.

5. Kim

Haben Sie ein gutes Gedächtnis? Wenn Sie glauben, daß Sie sich Dinge gut einprägen können, dann sollten Sie einmal das Kim-Spiel spielen. Eine Computerversion des alten Pfadfinderspiels.

6. Turf

4 Pferde sind am Start und Sie können Wetten abschließen. Ein packendes Finish gibt es fast immer. Der Computer sorgt für Rennplatzstimmung und Wettfieber.

7. High Noon

Wenn Sie Ihre Reaktionsschnelligkeit testen und verbessern wollen, dann ist High Noon genau das richtige Programm für Sie. Je schneller Sie drücken, desto mehr Punkte gibt es.

8. Torten

In eine Tortenschlacht a la Dick und Doof versetzt Sie dieses Programm. Wie viele Gegner können Sie ausschalten, bevor Sie völlig zugeschmiert sind?

9. U-Boot

Als Kommandant eines U-Boot-Jägers müssen Sie das U-Boot finden und vernichten, bevor es Sie vernichtet. Es ist gar nicht so einfach, obwohl Ihr Bordcomputer ein Sharp PC ist.

10. Frosch

8 Frösche und 9 Seerosenblätter, das ist die Ausgangsposition für das knifflige Strategiespiel. Wer einfach draufloshüpft, der wird viel zu viele Sprünge benötigen. Wieviele Sprünge brauchen Sie?

2.6.1 Henker

Henker ist ein Spiel, daß auf jedem Computersystem zu finden sein dürfte. Besser bekannt ist es unter dem Namen "Hangman". Der Computer wählt ein Wort aus den Datazeilen des Programms aus, und der Spieler muß es erraten. Es darf jeweils ein Buchstabe vom Spieler geraten werden und der Computer setzt den Buchstaben dann an den Stellen im Wort ein, an denen dieser Buchstabe vorkommt. Die Aufgabe scheint auf den ersten Blick nicht besonders schwierig zu sein, aber die zu erratenden Begriffe haben es in sich.

Wer der Meinung ist, diese Spiel sei ihm zu einfach und habe keinen Nutzen für ihn, dem habe ich 3 Vorschläge zu machen.

1. Versuchen Sie, nach jedem erratenen Begriff, diesen Begriff kurz zu definieren.
2. Versuchen Sie es mit dem Programm Rekneh. Sie werden feststellen, das dieses Programm höhere Ansprüche stellt.
3. Wenn Ihnen auch Rekneh zu leicht geworden ist, dann empfehle ich Ihnen, einfach neue Begriffe in die Datazeilen zu setzen.

Variablenliste:

A\$(1): Striche für noch nicht geratene Begriffe
 H\$(1): Zwischenspeicher
 S\$(1): Geratener Buchstabe
 Z\$(1): Das zu ratende Wort
 N: Nummer des ausgewählten Wortes

```

1 CLEAR:DIM Z$(1):DIM A$(1)
2 WAIT 150:DIM H$(1)
3 RESTORE
4 PRINT"   HENKER"
5 PRINT"  IM SPCS"
6 N=INT RND 25
7 FOR I=1 TO N
8 READ Z$
9 NEXT I
10 WAIT 300
11 A$(1)="-----"
12 A$(1)=LEFT$(A$(1),LEN Z$(1))
13 PRINT A$(1)
14 S$=INKEY$:IF S$="" THEN 14
15 H$(1)=""
16 R=0

```

Do not sale !

```
17 FOR I=1 TO LEN Z$(1)
18 IF S$=MID$(Z$(1),I,1) THEN 21
19 H$(1)=H$(1)+MID$(A$(1),I,1)
20 GOTO 23
21 H$(1)=H$(1)+S$
22 R=1
23 NEXT I
24 A$(1)=H$(1)
25 IF R>0 THEN 28
26 X=X+1
27 IF X>8 THEN 38
28 IF A$(1)=Z$(1) THEN 31
29 IF A$(1)<>S$ THEN 12
30 PRINT"ENDE"
31 PRINT A$(1)
32 WAIT 300
33 PRINT"BRAVO !!!!!"
34 WAIT 500
35 PRINT X;"-FEHLER"
36 WAIT 150
37 GOTO 50
38 PRINT X;"-FEHLER !!"
39 WAIT 500
40 PRINT"DU BIST GEHAENGT!"
41 WAIT 150
42 GOTO 50
43 DATA "FLOPPY","DISKETTE","DATABECKER","COMPUTER"
44 DATA "TASTATUR","SHARPPC","CASSETTE","JOYSTICK"
45 DATA "FLIPFLOP","PARITAET","MONITOR","HALBBYTE"
46 DATA "NIBBLE","ASSEMBLER","BASIC","FORTH"
47 DATA "PASCAL","FORTRAN","COBOL","COMPILER"
48 DATA "INTERPRETER","WINCHESTER","SNOBOL"
49 DATA "SMALLTALK","XENIX"
50 INPUT"NEUES SPIEL ";N$
51 IF N$="J" THEN 1
52 PRINT"TSCHUESS"
53 END
```


2.6.2 Jackpot

Einarmige Banditen bieten ein hohes Maß an Risiko und damit zugleich einen großen Nervenkitzel. Diesen Nervenkitzel können auch Sie jetzt überall haben. Das folgende Programm simuliert einen einarmigen Banditen. Das Programm hat sogar einen großen Vorteil gegenüber den echten Automaten. Hier kann man kein Geld verlieren, sondern höchstens seine Programmierkenntnisse erweitern.

Während des Spielablaufs hat man zusätzlich die Möglichkeit, auf den Spielverlauf Einfluß zu nehmen. Nach einer bestimmten Durchlaufzahl erscheint neben den rotierenden Anzeigen ein S, durch das die Möglichkeit angezeigt wird, das linke Feld wieder zu starten.

Ihr Startkapital beträgt 15 DM, die Gewinnquote ist sowohl abhängig von der Anzahl der gleichen Felder, als auch von den Buchstaben.

Gewinne:

3 gleiche Symbole = Jackpot
A-B-C = Status 1
2 gleiche im linken und mittleren Feld = Status 2
Ein A im linken Feld = Status 3

Variablenliste:

A\$: Linkes Feld
B\$: Mittleres Feld
C\$: Rechtes Feld
D\$: Startmöglichkeit
O\$: Tastaturabfrage
Z\$: Tastaturabfrage
A: Wert für linkes Feld
B: Wert für mittleres Feld

Do not sale !

C: Wert für rechtes Feld
G: Kapital
I: Schleifenvariable
W: Schaltvariable

```
1 CLEAR
2 WAIT 150
3 W=0
4 G=15
5 PRINT">> JACKPOT <<"
6 PRINT" > IM SPCS <"
7 PRINT"S---> START"
8 A$=INKEY$
9 IF A$="" THEN 8
10 IF A$<>"S" THEN 8
11 G=G-1
12 FOR I=1 TO 140
13 IF I=60 THEN LET W=1
14 IF W=1 THEN LET D$="S":GOTO 16
15 A=RND 7:A$=CHR$ A+64
16 IF I>80 THEN LET D$="":W=0:GOTO 18
17 B=RND 7:B$=CHR$ B+64
18 IF I>120 THEN 20
19 C=RND 7:C$=CHR$ C+64
20 GOTO 22
21 NEXT I
22 WAIT 20:PRINT">> ";A$;B$;C$;" <<";D$
23 O$=INKEY$
24 IF O$="S" THEN LET W=0:D$=""
25 IF I<140 THEN 27
26 GOSUB 41
27 IF I<140 THEN 21
28 WAIT 300
29 PRINT"GELD: ";G;"--> S"
30 IF G=0 THEN 32
31 GOTO 8
32 WAIT 300
33 PRINT"DU HAST ALLES"
34 PRINT"VERLOREN !!"
```

Do not sale !

```
35 PRINT"NEUES SPIEL:"
36 Z$=INKEY$
37 IF Z$="" THEN 36
38 IF Z$="J" THEN 1
39 PRINT"BYE BYE"
40 END
41 WAIT 300
42 IF A$=B$ AND B$=C$ THEN PRINT"JACKPOT":G=G+A*6:GOTO 46
43 IF A$="A" AND B$="B" AND C$="C" THEN PRINT"STATUS 1":G=G+15:GOTO 46
44 IF A$=B$ THEN PRINT"STATUS 2":G=G+A*3:GOTO 46
45 IF A$="A" THEN PRINT"STATUS 3":G=G+3
46 WAIT 10
47 RETURN
```

Dieses Programm läßt sich natürlich hervorragend als Grundlage für Jackpotspiele mit 5 Feldern nutzen. Allerdings ist diese Erweiterung mit einem hohen Arbeitsaufwand verbunden. Der Arbeitsaufwand entsteht hauptsächlich in der Auswertung der Ergebnisse, da hier neue und umfangreiche Staffeln erstellt werden müssen.

2.6.3 Zahlenschloß

Jedem dürfte bekannt sein, was ein Zahlenschloß ist und wie es funktioniert. Man kann es nur öffnen, wenn man die richtige Zahlenkombination wählt. Das Sicherungsprinzip beruht auf der Tatsache, daß es so viele Kombinationen gibt, daß es nur sehr schwer oder gar nicht möglich ist, dieses Schloß in einem realistischen Zeitraum zu knacken.

Dieses Programm simuliert ein verschlossenes Zahlenschloß, und Ihre Aufgabe ist es natürlich, dieses Schloß möglichst schnell zu knacken. Die verschlüsselte Zahl hat 5 Stellen. Zu Beginn des Programms haben Sie die Möglichkeit, die Obergrenze der einzelnen Stellen festzulegen. Wenn Sie beispielsweise die Zahl 6 als Obergrenze gewählt haben, dann liegt der Wert jeder Stelle zwischen 1 und 6 einschließlich.

Do not sale !

Nachdem Sie die Obergrenze festgelegt haben, können Sie daran gehen, die Kombination zu erraten. Natürlich gibt Ihnen der Rechner Hilfen. Wenn Sie die 5 Zahlen eingegeben haben, gibt der Rechner bei richtig geratenen Zahlen die Zahlen an, die richtig waren. Sollte nach der Eingabe Ihrer Kombination der Rechner eine 3 ausgeben, so ist die dritte Stelle Ihrer Kombination richtig gewesen.

Wenn Sie bei der Wertung über eine lapidare Floskel hinauskommen wollen, dann werden Sie sich ganz schön ins Zeug legen müssen. Aber Sie sollten sich auf keinen Fall entmutigen lassen.

Variablenliste:

A\$:	Antwort
A():	Zahlen von 1 bis 5
A:	Zufallszahl
B:	Stelle
C:	Stelle
D:	Stelle
E:	Stelle
F:	Stelle
I:	Schleifenvariable
J:	Schleifenvariable
X(I):	Inhalt der Kombination
Y:	Obergrenze
Z:	Zahl der Versuche

```
1 CLEAR
2 WAIT 150
3 PRINT"ZAHLENSCHLOSS"
4 PRINT">> IM SPCS <<"
5 INPUT"OBERGRENZE:";Y
6 IF Y<5 THEN 8
7 GOTO 9
8 PRINT"ZU KLEIN":GOTO 5
9 FOR I=1 TO 5
```

Do not sale !

```
10 A=RND Y
11 FOR J=1 TO 5
12 IF A=X(J) THEN 10
13 NEXT J
14 X(1)=A:NEXT I
15 A$(1)="1 ,"
16 A$(2)="2 ,"
17 A$(3)="3 ,"
18 A$(4)="4 ,"
19 A$(5)="5 ,"
20 PRINT "VERSUCH ":Z=Z+1
21 INPUT B,C,D,E,F
22 IF B+C+D+E+F=1 THEN PRINT X(1);X(2);X(3);X(4);X(5):GOTO 20
23 IF B<>X(1) THEN 29
24 IF C<>X(2) THEN 29
25 IF D<>X(3) THEN 29
26 IF E<>X(4) THEN 29
27 IF F<>X(5) THEN 29
28 GOTO 38
29 FOR I=1 TO 5
30 IF B=X(I) THEN PRINT A$(I);:GOTO 35
31 IF C=X(I) THEN PRINT A$(I);:GOTO 35
32 IF D=X(I) THEN PRINT A$(I);:GOTO 35
33 IF E=X(I) THEN PRINT A$(I);:GOTO 35
34 IF F=X(I) THEN PRINT A$(I):GOTO 35
35 NEXT I
36 GOTO 20
37 END
38 PRINT"ICH GRATULIERE!"
39 PRINT Z;" VERSUCHE"
40 IF Z>7 THEN PRINT"SCHLAPPE LEISTUNG"
41 INPUT"NEUES SPIEL: ";A$
42 IF LEFT$(A$,1)="J" THEN 1
43 END
```

Die einfachste Art, dieses Programm noch schwieriger zu gestalten, dürfte die Erweiterung der Anzahl der Stellen sein.

Do not sale !

Wenn Sie sich die Aufgabe mit 6 oder noch mehr Stellen auferlegen, dann werden Sie sicher mit dem Schwierigkeitsgrad zufrieden sein.

2.6.4 Rekneh

Der Name dieses Programmes wird Ihnen auf den ersten Blick wahrscheinlich gar nichts sagen. Diese Tatsache wird mindestens solange Bestand haben, bis Sie das Wort von rechts nach links lesen.

Sehr aufregend ist die Sache auch sicher dann noch nicht. Aufregend wird die Sache erst, wenn Sie Rekneh spielen ohne sich vorher über den Hintergrund dieses Programmes informiert zu haben. Rekneh ist gewissermaßen eine spezielle Hangman-version, die mehr als eine normale Hangmanversion von Ihnen fordert.

Sie müssen, genau wie bei Henker, Worte erraten. Die Schwierigkeit liegt nun in der Tatsache, daß das zu erratende Wort von rechts nach links geschrieben wird. Wenn Sie also das Wort FLOPPY erraten sollen, so steht auf Ihrem Display "YPPOLF". Dieser Umstand erschwert die Aufgabe erheblich. Einen kleinen Vorteil haben natürlich die Leute, die schon das Programm "Von Rechts Nach Links" eingetippt haben und dadurch schon Übung im Rückwärtssprechen haben.

Variablenliste:

A\$(1):	Leerdisplay
B\$(1):	Das geratene Wort
Z\$(1):	Das zu ratende Wort
K\$:	Tastatureingabe
I:	Schleifenvariable
R:	Schaltvariable
N:	Nummer des gewählten Wortes
T:	Länge des zu ratenden Wortes
X:	Anzahl der Fehler

Einen Vorschlag kann ich Ihnen auch zu diesem Programm machen. Sollte Ihnen dies Programm zu einfach geworden sein, dann ändern Sie doch einfach die Begriffe in den Datenzeilen.

```
1 CLEAR
2 WAIT 150
3 DIM A$(1):DIM B$(1)
4 DIM Z$(1)
5 RESTORE
6 PRINT"REKNEH"
7 PRINT"SPEZIAL"
8 PRINT" IM SPCS"
9 PRINT"VIEL SPAB !!"
10 N=INT RND 25
11 FOR I=1 TO N
12 READ Z$(1)
13 T=LEN Z$(1)
14 FOR I=T TO 1 STEP -1
15 K$=K$+MID$(Z$(1),I,1)
16 NEXT I
17 Z$(1)=K$
18 WAIT 200
19 A$(1)="-----"
20 A$(1)=LEFT$(A$(1),LEN Z$(1))
21 PRINT A$(1)
22 S$=INKEY$:IF S$="" THEN 22
23 B$(1)=""
24 R=0
25 FOR I=1 TO LEN Z$(1)
26 IF S$=MID$(Z$(1),I,1) THEN 29
27 B$(1)=B$(1)+MID$(A$(1),I,1)
28 GOTO 31
29 B$(1)=B$(1)+S$
30 R=1
31 NEXT I
32 A$(1)=B$(1)
```

Do not sale !

```
33 IF R>0 THEN 36
34 X=X+1
35 IF X>8 THEN 46
36 IF A$(1)=Z$(1) THEN 39
37 IF A$(1)<>S$ THEN 20
38 PRINT"ENDE"
39 PRINT A$(1)
40 WAIT 300
41 PRINT"BRAVO !!!!!"
42 WAIT 500
43 PRINT X;"-FEHLER"
44 WAIT 150
45 GOTO 57
46 PRINT X;"-FEHLER !!"
47 PRINT"PECH GEHABT !"
48 PRINT"DU BIST GEHAENGT!"
49 WAIT 150
50 GOTO 57
51 DATA "FLOPPY","DISKETTE","DATABECKER","COMPUTER"
52 DATA "TASTATUR","SHARPPC","CASSETTE","JOYSTICK"
53 DATA "FLIPFLOP","PARITAET","MONITOR","HALBBYTE"
54 DATA "NIBBLE","ASSEMBLER","BASIC","PASCAL"
55 DATA "FORTRAN","COBOL","COMPILER","INTERPRETER"
56 DATA "WINCHESTER","SNOBOL","SMALLTALK","XENIX"
57 INPUT"NEUES SPIEL ";N$
58 IF N$="J" THEN 1
59 PRINT"TSCHUESS"
60 END
```

2.6.5 Das Kim-Spiel

Jeder (ehemalige) Pfadfinder wird sich gern an dieses Spiel erinnern. Im Zeitalter der Pocket Computer kann man dieses traditionsreiche Spiel mit Hilfe des Computers spielen.

Eine Zeichenkette mit maximal 16 Zeichen wird über den Zufallszahlengenerator erzeugt. Man gibt die Zeit ein, die man für das Einprägen der Zeichenkette benötigt. Danach werden die Zeichen mit Hilfe des Zufallszahlengenerators gemischt.

Do not sale !

Die gemischte Zeichenkette wird dann ausgegeben, und Sie müssen entscheiden, ob die Stellung des einzelnen Zeichens richtig ist, oder nicht. Wenn Sie der Ansicht sind, daß ein Zeichen an der falschen Stelle steht, dann müssen Sie auch das Zeichen angeben, das dort gestanden hat.

Alles in allem ist das Kim-Spiel in der Computerversion genauso unterhaltsam, wie das, was im Lager gespielt wurde.

Variablenliste:

A\$():	Speicher
B\$():	Speicher
C\$():	Speicher
D\$():	Speicher
E\$():	Speicher
I:	Schleifenvariable
J:	Schleifenvariable
L:	Schleifenvariable
M:	Zahl der Fehler
N:	Schleifenvariable
S:	Displayzeit
U:	Zwischenspeicher
X:	Zeichenwert
Y:	Zufallswert

```
10 CLEAR
20 WAIT 150
30 PRINT"KIM SPIEL"
40 PRINT"IM SPCS"
50 INPUT"ZEICHENZAHL: ";J
60 DIM A$(J)
70 DIM C$(J)
80 DIM E$(J)
90 DIM B$(0)
100 DIM D$(0)
110 B$(0)=""
```

Do not sale !

```
120 D$(0)=""
130 M=0
140 FOR I=0 TO J-1
150 X=RND 62
160 X=X+33
170 A$(I)=CHR$ X
180 B$(0)=B$(0)+A$(I)
190 NEXT I
200 S=20
210 PRINT"BIITE MERKEN !"
220 S=S*65
230 WAIT S
240 PRINT B$(0)
250 FOR I=0 TO J-1
260 E$(I)=A$(I)
270 NEXT I
280 FOR N=0 TO J-1
290 U=J-M
300 Y=RND U
310 C$(N)=E$(Y-1)
320 IF Y=J THEN 360
330 FOR Z=Y TO J-1
340 E$(Z-1)=E$(Z)
350 NEXT Z
360 D$(0)=D$(0)+C$(N)
370 NEXT N
380 WAIT 120
390 PRINT"BITTE MERKEN !"
400 WAIT S
410 PRINT D$(0)
420 E=5
430 E=E*65:WAIT E
440 FOR I=1 TO J
450 PRINT I;" ZEICHEN ";C$(I-1);"?"
460 F$=""
470 F$=INKEY$
480 IF F$="J" THEN 530
490 IF F$="N" THEN 670
500 GOTO 460
510 NEXT I
```

Do not sale !

```
520 GOTO 730
530 IF A$(I-1) <> C$(I-1) THEN 560
540 PRINT "RICHTIG !"
550 GOTO 510
560 PRINT "FALSCH !"
570 M=M+1
580 C$(I-1)=A$(I-1)
590 D$(O)=" "
600 FOR Z=0 TO J-1
610 D$(O)=D$(O)+C$(Z)
620 NEXT Z
630 WAIT 120
640 PRINT M;" FEHLER !"
650 PRINT D$(O)
660 GOTO 510
670 IF C$(I-1)=A$(I-1) THEN 560
680 WAIT 120
690 INPUT "WAS IST RICHTIG ?";E$(I-1)
700 IF E$(I-1)=A$(I-1) THEN 720
710 GOTO 560
720 PRINT "RICHTIG !":GOTO 580
730 WAIT 300
740 G=J/3
750 S=2*G
760 PRINT M;" FEHLER !"
770 END
```

Eine pikante Möglichkeit, dieses Programm noch interessanter zu machen, ist die Möglichkeit, zusätzliche Zeichen neu zu definieren. Diese Erweiterung ist natürlich nicht im Rahmen des SPCS möglich.

2.6.6 Turf

Pferderennen ist eine aufregende Angelegenheit. Wer allerdings dem Pferdewetten frönt, der weiß wie teuer dieser Spaß ist. Mit diesem Programm holen Sie sich die Rennbahn direkt auf das Display Ihres PC. Bei jedem Rennen laufen 4 Pferde gegenein-

Do not sale !

ander, und Sie haben vor dem Rennen die Gelegenheit von Ihrem Kapital (300 DM) eine Summe auf eines der Pferde zu setzen.

Die Distanz, die die Pferde zurückzulegen haben, wird in Längeneinheiten angegeben. Das Pferd, das zuerst 50 Längeneinheiten zurückgelegt hat, hat gewonnen. Das Programm ist so gestaltet, daß ein packendes Finish vorprogrammiert ist.

Aber seien Sie vorsichtig mit Ihren Einsätzen, denn der Computer gewährt keinen Kredit. Sie sind also gezwungen, Ihr Kapital sorgsam einzusetzen.

Achtung: Der CURSOR-Befehl in der Zeile 22 ist nicht auf allen Sharp PC vorhanden. Diese Zeile darf nur eingesetzt werden, wenn der Rechner den CURSOR-Befehl verarbeiten kann.

Variablenliste:

M\$:	Antwort
Q\$:	Zwischenspeicher
A(x):	Distanz der Pferde
E:	Einsatz
G:	Kapital
P:	Gesetzte Pferd Nummer
X:	Nummer des Pferdes

```
1 CLEAR
2 WAIT 150
3 DIM A(4)
4 PRINT">> TURF <<"
5 PRINT">IM SPCS<"
6 G=300
7 IF G=0 THEN 35
8 X=0
9 A(1)=0
10 A(2)=0
11 A(3)=0
```

Do not sale !

```

12 A(4)=0
13 PRINT"GELD:";G
14 INPUT"PFERD 1,2,3,4:";P
15 IF P<1 OR P>4 THEN 14
16 INPUT"IHR EINSATZ:";E
17 IF G<E THEN PRINT"KEIN KREDIT !":GOTO 16
18 G=G-E
19 X=RND 4
20 WAIT 40
21 A(X)=A(X)+1
22 CURSOR 0:'NUR SETZEN WENN VORHANDEN
23 PRINT A(1);" ";A(2);" ";A(3);" ";A(4);"      "
24 IF A(X)=50 THEN 26
25 GOTO 19
26 PRINT"AUSWERTUNG !"
27 WAIT 40
28 IF A(P)<>50 THEN WAIT 150:PRINT"VERLOREN!":GOTO 7
29 E=E+(E/2+RND 90)
30 G=G+E
31 BEEP 10
32 WAIT 150
33 PRINT"GEWONNEN !!"
34 GOTO 7
35 WAIT 150
36 PRINT"DU HAST ALLES"
37 PRINT"VERLOREN !!"
38 INPUT"NEUES SPIEL:";M$
39 Q$=LEFT$(M$,1)
40 IF Q$="J" THEN 1
41 PRINT"TSCHUESS !"
42 END

```

Wenn Sie dieses Programm erweitern wollen, dann sollten Sie versuchen, eine Sieg- und Platzwette einzubauen. Eine weitere Möglichkeit, dieses Programm zu verändern, ist die Gewinnausschüttung. Hier bietet sich genügend Möglichkeit, den Schwierigkeitsgrad des Programms zu verändern.

Do not sale !

2.6.7 High Noon

Wer den berühmten Western High Noon gesehen hat, der weiß genau warum man reaktionsschnell sein muß. Aber nicht nur im Western, sondern auch im täglichen Leben des modernen Menschen ist Reaktionsschnelligkeit von Nutzen. Das Paradebeispiel für diese Tatsache ist das Autofahren. Wer hier schlechte Reaktionszeiten aufweist, dessen Gesundheit ist mehr gefährdet, als die des reaktionsschnellen Nachbarn. Somit ist dieses Programm ein echter Beitrag zur Verbesserung und Sicherung Ihrer Gesundheit.

Zu Beginn dieses Programms erscheint nach der Anfangsmeldung eine Aufforderung, das eigentliche Programm durch "S" zu starten. Daraufhin erscheint eine Zahl auf dem Display. Ihre Aufgabe ist es, möglichst schnell auf die entsprechende Taste Ihrer Tastatur zu drücken. Bei den erscheinenden Zahlen handelt es sich um die Zahlen 1, 3, 7 und 9. Diese Zahlen wurden ausgewählt, da sie sich durch ihre Stellung auf der Tastatur besonders dafür eignen.

Wenn die Zahl im Display erscheint, sind die aktuellen Punkte auf 100 gesetzt. Danach wird die Punktzahl recht schnell reduziert, bis Sie die richtige Zahl gedrückt haben. Da jedes Spiel 10 Durchgänge hat, sind also 1000 Punkte möglich. Die 1000 Punkte sind allerdings nur theoretisch zu erreichen. Punktzahlen, die über 980 liegen, sind schon als sehr gut einzustufen.

Variablenliste:

A\$(1):	Zahl 1
A\$(2):	Zahl 3
A\$(3):	Zahl 7
A\$(4):	Zahl 9
Z\$:	Tastaturabfrage
G:	Gesamtpunkte

N: Schleifenvariable
P: Aktuelle Punktzahl
Q: Anzahl der Durchgänge
X: Zufallszahl

```
1 CLEAR
2 WAIT 300
3 PRINT"HIGH NOON"
4 PRINT"IM SPCS"
5 DIM A$(4)
6 A$(1)="1"
7 A$(2)="3"
8 A$(3)="7"
9 A$(4)="9"
10 WAIT 20
11 PRINT"S--> START"
12 Z$=INKEY$
13 IF Z$="" THEN 12
14 IF Z$<>"S" THEN 11
15 PRINT""
16 X=RND 4
17 P=100
18 CURSOR RND 12:REM Zeile nur setzen, wenn CURSOR vorhanden
19 PRINT A$(X)
20 Z$=INKEY$
21 P=P-1
22 IF Z$=A$(X) THEN 25
23 FOR N=1 TO 50:NEXT N
24 GOTO 20
25 G=G+P
26 WAIT 80
27 PRINT P;" PUNKTE"
28 WAIT 20
29 PRINT""
30 Z$=""
31 Q=Q+1
32 IF Q=10 THEN 34
33 GOTO 16
```

Do not sale !

```
34 WAIT 300
35 PRINT"DU HAST VON"
36 PRINT"1000 PUNKTEN"
37 PRINT G;" PUNKTE"
38 END
```

Wenn Ihnen das Programm in dieser Form zu einfach geworden ist, dann sollten Sie das Programm so modifizieren, daß 6 verschiedene Zahlen im Display erscheinen können. Dies dürfte die leichteste Möglichkeit sein, das Level dieses Programms zu heben.

2.6.8 Torten

Tortenschlacht wäre eine genauere Beschreibung für dieses klebrige Spiel. Der Computer versetzt Sie direkt in eine gnadenlose Tortenschlacht. Natürlich haben Sie Ihren Gegnern gegenüber einen entscheidenden Vorteil, denn Sie haben einen Pocket Computer. Der Computer hilft Ihnen, den Gegner durch einen gezielten Tortenwurf auszuschalten.

Der Rechner gibt Ihnen zunächst die Entfernung des zu treffenden Gegners an. Ihre Aufgabe ist es nun, durch Angabe des Abwurfwinkels und der Abwurfwucht, die Torte in das Gesicht des Opfers zu befördern.

Für jeden Gegner haben Sie 4 Versuche. Sollten Sie den Gegner allerdings zum vierten Mal verfehlt haben, werden Sie von einer Torte getroffen und Ihre Sicht verringert sich merklich.

Variablenliste:

M\$:	Antwort
N\$:	Antwort
A:	Abwurfwinkel
G:	Erdanziehung
K:	Tempo des Abwurfes

Do not sale !

R: Sicht
V: Anzahl der Versuche
W: Wurfweite
X: Entfernung des Opfers

```
1 "A":REM Label kann weggelassen werden
2 CLEAR
3 WAIT 150
4 PRINT"TORTEN !!!"
5 PRINT"ODER....."
6 PRINT"SUESSES IM SPCS"
7 G=9.81
8 R=3
9 X=RND 85
10 RANDOM
11 IF X<10 THEN 9
12 V=0
13 X=INT X
14 PRINT"OPFER IN";X;"M"
15 V=V+1
16 PRINT"VERSUCH:";V
17 INPUT"ABWURFWINKEL:";A
18 IF A<1 OR A>180 THEN 17
19 IF A=90 THEN GOTO 32
20 INPUT"TEMPO:";K
21 IF K<1 OR K>10 THEN GOTO 150
22 IF A>90 THEN GOSUB 37
23 W=K^2*(SIN(2)*A)/G
24 W=INT(W)
25 PRINT"WEITE=";W
26 IF W=X THEN 49
27 PRINT"DU HAST DEIN"
28 PRINT"DEIN OPFER"
29 PRINT"UM";W-X;"M VERFEHLT"
30 IF V=4 THEN 42
31 GOTO 11
32 PRINT"DU HAST DICH"
33 PRINT"SELBST !!!"
34 PRINT"BEKLECKERT!!"
```

Do not sale !

```
35 R=R-1
36 GOTO 11
37 PRINT"IST DAS DER"
38 PRINT"RICHTIGE WEG?"
39 INPUT M$
40 IF M$="J" THEN PRINT"IDIOT !!!"
41 RETURN
42 PRINT"DU BIST.."
43 PRINT"BEKLECKERT"
44 PRINT"WORDEN !!!"
45 R=R-1
46 PRINT"SICHT=";R
47 IF R=3 THEN 54
48 GOTO 11
49 PRINT"BRAVO DU HAST"
50 PRINT"ES GETROFFEN!!"
51 T=T+1
52 PRINT"TREFFER=";T
53 GOTO 11
54 PRINT"DU BIST VOELLIG"
55 PRINT"VERKLEBT !!!"
56 PRINT"DIE SCHLACHT"
57 PRINT"IST AUS !!!"
58 INPUT"NEUES SPIEL=";N$
59 IF N$="J" THEN 1
60 END
```

2.6.9 U-Boot

Sie sind der Kapitän eines Kreuzers, dessen Aufgabe es ist, U-Boote aufzuspüren und zu vernichten. Diese Aufgabe sollen Sie mit Hilfe von Wasserbomben und Ihres Sharp PC-Bordrechners lösen. Zur Orientierung dienen Ihnen 3 Koordinaten.

Die Koordinate X reicht von 1 (Westen) bis 20 (Osten).
Die Koordinate Y reicht von 1 (Sueden) bis 20 (Norden).
Die Koordinate Z reicht von 1 (Oberfläche) bis 20 (Tief).

Die Eingaben müssen zwischen 1 und 20 liegen und ganze Werte sein. Nach jedem Versuch gibt Ihnen der Bordcomputer an, ob Sie z.B. zu oestlich, zu noerdlich oder zu flach geschossen haben.

Bis jetzt war ja alles ganz harmlos für Sie, das wird jetzt anders. Sie haben maximal 4 Versuche, das U-Boot zu treffen. Wenn Sie es nach dem vierten nicht getroffen haben, werden Sie vom U-Boot torpediert.

Variablenliste:

M\$:	Abfrage Angaben wiederholen
N\$:	Abfrage neues Spiel
Q\$:	Zwischenspeicher
W\$:	Zwischenspeicher
A:	Versuch für X
B:	Versuch für Y
C:	Versuch für Z
V:	Anzahl der Versuche
X:	Koordinate für U-Boot
Y:	Koordinate für U-Boot
Z:	Koordinate für U-Boot

```
1 CLEAR
2 WAIT 300
3 PRINT"U-BOOT"
4 X=(RND 19)+1
5 Y=(RND 19)+1
6 Z=(RND 19)+1
7 IF A=X THEN 10
8 PRINT A;
9 INPUT"KOORDINATE X";A
10 IF B=Y THEN 13
11 PRINT B;
12 INPUT"KOORDINATE Y";B
13 IF C=Z THEN 16
14 PRINT C;
15 INPUT"KOORDINATE Z:";C
```

Do not sale !

```
16 IF A>20 OR A<1 THEN 35
17 IF B>20 OR B<1 THEN 35
18 IF C>20 OR C<1 THEN 35
19 IF A<X THEN PRINT"X-ZU WESTLICH"
20 IF A>X THEN PRINT"X-ZU OESTLICH"
21 IF A=X THEN PRINT"X- OK":BEEP 2
22 IF B<Y THEN PRINT"Y-ZU SUEDLICH"
23 IF B>Y THEN PRINT"Y-ZU NOERDLICH"
24 IF B=Y THEN PRINT"Y- OK":BEEP 2
25 IF C<Z THEN PRINT"Z-ZU FLACH"
26 IF C>Z THEN PRINT"Z-ZU TIEF"
27 IF C=Z THEN PRINT"Z- OK":BEEP 2
28 INPUT"ANGABEN NOCHMAL";M$
29 Q$=LEFT$(M$,1)
30 IF Q$="J" THEN 19
31 IF C=Z AND B=Y AND A=X THEN 37
32 V=V+1
33 IF V=5 THEN 44
34 GOTO 7
35 PRINT"FEHLER !!!"
36 GOTO 7
37 PRINT"U-BOOT VERSENKT"
38 PRINT"BRAVO !!!"
39 INPUT"NEUES SPIEL:";N$
40 W$=LEFT$(N$,1)
41 IF W$="J" THEN 1
42 PRINT"ENDE DER MISSION"
43 END
44 PRINT"SIE WURDEN"
45 PRINT"TORPEDIERT!!!"
46 PRINT"X-";X;"Y-";Y;"Z-";Z
47 PRINT"SO LONG BOY"
48 GOTO 39
```

Vorschlag:

Sollten Ihnen die Aufgabe zu leicht fallen, dann sollten Sie die Zahl der Versuche in Zeile 33 verringern. Eine weitere Möglichkeit, das Spiel noch interessanter zu machen, wäre ein

Do not sale !

bewegliches U-Boot. Das U-Boot sollte sich nach Möglichkeit alle 2 Versuche um 1 Zähler bewegen.

2.6.10 Frosch

Keine Angst! Es handelt sich bei diesem Programm nicht um Versuch, eine Frogger Version auf einem einzeiligen Display unterzubringen. Vielmehr ist Frosch ein sehr interessantes Spiel, bei dem es auf das strategische Denkvermögen ankommt. Es ist ratsam, sich vorher ein System zu überlegen, da die Zahl der Züge sonst zu groß wird.

Folgende Vorstellung liegt diesem Spiel zugrunde:

Auf einem kleinen See sind 9 Seerosenblätter in einer Reihe nebeneinander. Auf den 4 linken Blättern sitzen 4 dunkelgrüne Frösche (A), und auf den 4 rechten Blättern sitzen 4 hellgrüne Frösche (H). Das Blatt in der Mitte ist frei. Der Abstand der Blätter ist so groß, daß ein Frosch maximal 2 Blätter weiter springen kann. Natürlich dürfen nicht 2 Frösche auf einem Blatt sitzen, da das Blatt sonst untergeht.

Ihre Aufgabe ist es nun, die Frösche so zu dirigieren, daß die hellgrünen den Platz mit den dunkelgrünen tauschen. Dabei sollen möglichst wenig Sprünge benötigt werden. Das Spiel ist beendet, wenn alle hellgrünen Frösche mit den dunkelgrünen Fröschen die Plätze getauscht haben.

Die Eingabe der Züge erfolgt durch Angabe der betroffenen Blattnummern:

AAAA HHHH - Frösche
123456789 - Blattnummern

Beispiel: Die Eingabe 3 (Enter) 5 (Enter) bewirkt, daß der Frosch von Blatt 3 auf Blatt 5 (in der Mitte) springt.

Viel Spaß!

Do not sale !

Variablenliste:

B\$(I): Blätter 1-9
C\$: Tastaturabfrage
D\$: Tastaturabfrage
H\$: Zwischenspeicher
F: Numerischer Wert von C\$
G: Numerischer Wert von D\$
I: Schleifenvariable
X: Anzahl der Sprünge

```
1 CLEAR
2 WAIT 300
3 PRINT">> FROSCH <<"
4 PRINT">> IM SPCS <<"
5 DIM B$(9)
6 FOR I=1 TO 9
7 READ H$
8 B$(I)=H$
9 NEXT I
10 RESTORE
11 FOR I=1 TO 9
12 PRINT B$(I);
13 NEXT I
14 PRINT " ";X
15 CURSOR 0:REM Diese Zeile nur einsetzen wenn CURSOR vorhanden
16 C$=INKEY$: IF C$="" THEN 16
17 F=VAL C$
18 IF F<1 OR F>9 THEN 32
19 FOR I=1 TO 200:NEXT I
20 D$=INKEY$:IF D$="" THEN 20
21 G=VAL D$
22 IF G<1 OR G>9 THEN 32
23 IF ABS(F-G)>2 THEN 32
24 IF B$(G)<>" " THEN 32
25 IF B$(F)=" " THEN 32
26 B$(G)=B$(F)
```

```
27 B$(F)=" "  
28 X=X+1  
29 FOR I=1 TO 9  
30 IF B$(I)=MID$("HHHH AAAA",I,1) THEN NEXT I:WAIT 500:GOTO 35  
31 GOTO 11  
32 BEEP 2  
33 PRINT"FEHLER !!!"  
34 GOTO 11  
35 PRINT"HHHH AAAA"  
36 IF X>40 THEN 38  
37 PRINT"BRAVO !!!"  
38 PRINT"DU HAST ES"  
39 PRINT"GESCHAFFT !!!"  
40 PRINT"DU HAST"  
41 PRINT X;" ZUEGE"  
42 PRINT"BENOETIGT"  
43 INPUT"NEUES SPIEL";A$  
44 W$=LEFT$(A$,1)  
45 IF W$="J" THEN 6  
46 PRINT"BYE BYE !"  
47 END  
48 DATA "A","A","A","A"," ","H","H","H","H"
```

Als Vorschlag zur Erweiterung des Programms möchte ich folgendes empfehlen:

Versuchen Sie die Anzahl der Blätter und der Frösche zu erhöhen. Gleichzeitig kann man in der Mitte 2 Blätter freilassen.

Do not sale !

3. Der Sharp-PC-Standard

Do not sale !

Do not sale !

3.1 Das Standard-BASIC

In diesem Kapitel werden die einzelnen BASIC-Befehle nach Funktionsbereichen aufgeführt und kurz erklärt. Dieses Kapitel soll Ihnen als Gedächtnisstütze dienen, um Ihnen das Programmieren auf Ihrem Sharp PC so einfach wie möglich zu machen. Im Anhang dieses Buches ist zusätzlich noch ein Verzeichnis der BASIC-Befehle, dem Sie entnehmen können, welcher Befehl auf welcher Seite zu finden ist.

3.1.1 Numerische Funktionen

ABS x

Errechnet den Absolut-Betrag des angegebenen Arguments.

$$\text{ABS } 5.5 = 5$$

$$\text{ABS } -5.5 = 5$$

ACS x

Errechnet den Arcussinus vom angegebenen Argument. Arcussinus ist die inverse Funktion zur Cosinusfunktion. Somit ist das Ergebnis der Winkel des gegebenen Wertes. Achtung: Das Ergebnis ist abhängig vom Winkelmodus des Rechners (DEG, RAD, GRAD).

$$\text{DEG-Modus ACS } 0.5 = 60$$

$$\text{RAD-Modus ACS } 0.5 = 1.047197551$$

$$\text{GRAD-Modus ACS } 0.5 = 60$$

ASN x

ASN berechnet den Arcussinus vom angegebenen Argument. Arcussinus ist die inverse Funktion zur Sinusfunktion. Auch hier ist das Ergebnis abhängig vom Winkelmodus des Rechners (DEG, RAD, GRAD).

DEG-Modus ASN 0.5 = 30

RAD-Modus ASN 0.5 = 5.235987756 E-01

GRAD-Modus ASN 0.5 = 30

ATN x

Errechnet den Arcustangens vom angegebenen Argument. Arcustangens ist die inverse Funktion zur Tangensfunktion. Wie bei allen trigonometrischen Funktionen, so ist auch hier das Ergebnis abhängig vom Winkelmodus des Rechners.

DEG-Modus ATN 1 = 45

RAD-Modus ATN 1 = 7.853984634 E-01

GRAD-Modus ATN 1 = 45

COS x

COS berechnet den Cosinus vom Winkelargument x. Auch bei der Cosinusfunktion ist das Ergebnis vom Winkelmodus des Rechners.

DEG-Modus COS 60 = 0.5

RAD-Modus COS 60 = -9.524129803 E-01

GRAD-Modus COS 60 = 0.5

DEG x

Die DEG-Funktion ist die Umkehrfunktion zur DMS-Funktion. Grad/Minute und Sekunde werden in dezimaler Form dargestellt. Der ganzzahlige Teil des Wertes wird als Grad(Stunde), die ersten beiden Dezimalstellen werden als Minuten und die dritte und vierte als Sekunden angesehen.

DEG 1.2010 = 1.336111111

Folglich sind 1 Stunde, 20 Minuten und 10 Sekunden = 1.33611111 Stunden

Do not sale !

DMS x

Wie schon oben erläutert, ist die DMS-Funktion die inverse Funktion der DEG-Funktion. Sie errechnet aus dezimalen Argumenten Stunden, Minuten und Sekunden.

$$\text{DMS } 1.336111111 = 1.20099999$$

Folglich sind 1.336111111 Stunden = 1 Stunde, 20 Minuten und 10 Sekunden (gerundet).

EXP x

Errechnet die Potenzen der Zahl e (= 12.718181828). Dieser Wert ist die Basis des natürlichen Logarithmus.

$$\begin{aligned} \text{Ln } 10 &= 2.302585093 \\ \text{EXP } 2.302585093 &= 10 \end{aligned}$$

INT x

Diese Funktion berechnet den ganzzahligen Anteil des Wertes x.

$$\text{INT } 10.321 = 10$$

Ln x

Berechnet den natürlichen Logarithmus (Basis e) von x.

$$\text{Ln } 100 = 4.605170185$$

LOG x

Errechnet den Logarithmus zur Basis 10 des Wertes.

$$\text{LOG } 100 = 2$$

PI

Die Kreiszahl PI ist eine numerische Pseudovvariable.

$$\text{PI} = 3.141592654$$

RND x

Durch die RND-Funktion wird eine Pseudozufallszahl errechnet. Ist x kleiner 1 und größer 0, so ist der errechnete Wert kleiner 1 und größer oder gleich 0. Wenn x eine ganze Zahl ist und größer oder gleich 1, so ist das Ergebnis größer oder gleich 1 und kleiner oder gleich x. Ist x keine ganze Zahl, so ist das Ergebnis größer oder gleich 1 und kleiner oder gleich der nächstgrößeren Zahl zu x.

$$\text{RND } 0.5 = \text{Ergebnis} > \text{gleich } 0 \text{ und } < 1$$

$$\text{RND } 10 = \text{Ergebnis zwischen } 1 \text{ und } 10$$

$$\text{RND } 9.3 = \text{Ergebnis zwischen } 1 \text{ und } 10$$

Die Zahlenfolge, die erzeugt wird, ist nach dem Einschalten des Rechners immer gleich, da die Basiszahl im Rechner fest implementiert ist. Will man diese Basiszahl verändern, so muß man sich des RANDOM-Befehles bedienen.

SGN x

Die Signum (Vorzeichen) Funktion berechnet das Vorzeichen einer Zahl. Ist x eine positive Zahl, so ist das Ergebnis 1. Ist x eine negative Zahl, so ist das Ergebnis -1. Sollte x=0 sein, so ist auch das Ergebnis 0.

Do not sale !

SGN -10 = -1
SGN 110 = 1
SGN 0 = 0

SIN x

Die Sinusfunktion errechnet den Sinus des Winkelargumentes *x*. Wie bei den anderen trigonometrischen Funktionen, so ist auch hier das Ergebnis abhängig vom Winkelmodus des Rechners (DEG, RAD, GRAD).

DEG-Modus SIN 30 = 0.5
RAD-Modus SIN 30 = -9.880316241 E-01
GRAD-Modus SIN 30 = 0.5

SQR x

Die SQR-Funktion berechnet die Quadratwurzel von *x*.

SQR 9 = 3

TAN x

Die Tangens-Funktion errechnet den Tangens des Winkelargumentes *x*. Das Ergebnis ist abhängig vom Winkelmodus des Rechners (DEG, RAD, GRAD).

3.1.2 Stringfunktionen

Ein String ist eine Reihe von alphanumerischen- und Sonderzeichen. String-Funktionen sind eine Reihe von Operationen, mit deren Hilfe man Strings manipulieren kann. Während man bei vielen BASIC-Dialekten das Argument der Funktion in Klammern setzen muß, so ist dies im SPCS nicht nötig. Klammern müssen nur dann gesetzt werden, wenn es notwendig ist,

das Funktionsargument deutlich von den anderen Werten zu trennen. Achtung! - Stringausdrücke mit zwei oder drei Ausdrücken benötigen immer Klammern.

ASC "A"

Die ASC-Funktion errechnet den Zeichen-Code des ersten Zeichens im String. Sharp PC arbeiten mit dem ASCII-Code. ASCII = American Standart Code for Information Interchange.

ASC "Z" = 90

ASC "8" = 56

ASC "!" = 33

CHR\$ 90

Die CHR\$-Funktion ist die Umkehrfunktion der ASC-Funktion. CHR\$ errechnet das ASCII-Zeichen des gegebenen Wertes.

CHR\$ 90 = Z

CHR\$ 56 = 8

CHR\$ 33 = !

Achtung: Der im SPCS darstellbare Zeichenbereich liegt zwischen 32 und 252. Andere Werte haben eine Fehlermeldung zur Folge.

INKEY\$

Die INKEY\$-Funktion prüft, ob und welche Taste gedrückt wurde. Dabei weist die INKEY\$-Funktion einem String das Zeichen der gedrückten Taste zu. Bei der Anwendung muß beachtet werden, daß die INKEY\$-Funktion nur im Augenblick der Bearbeitung des eigentlichen Befehls die Tastatur abgefragt wird. Das Programm hält also nicht wie beim INPUT-Befehl an, um auf eine Eingabe zu warten, sondern läuft sofort weiter.

Do not sale !

```
10 WAIT 150
20 PRINT" A ODER B"
30 AS=INKEYS
40 IF AS="A" THEN END
50 IF AS="B" THEN PRINT"HALLO"
```

Das Programm wird so, wie es hier abgedruckt ist, seinen Dienst nicht erfüllen, da die Chance, die Taste im richtigen Augenblick zu treffen sehr gering ist.

```
10 WAIT 150
20 PRINT" A ODER B"
30 AS=INKEYS: IF AS="" THEN 30
40 IF AS="J" THEN END
50 IF AS="B" THEN PRINT"HALLO"
```

Nun wird das Programm einwandfrei arbeiten, denn es wartet so lange, bis eine Taste gedrückt wurde.

LEFT\$ ("STRING",x)

Diese Funktion erstellt einen Teilstring der Länge x aus dem gegebenen String. Von links (engl.-left) beginnend werden x Zeichen herausgenommen.

```
LEFT$ ("SHARP",2) = "SH"
LEFT$ ("COMPUTER",5) = "COMPU"
```

RIGHT\$ ("STRING",x)

Die RIGHT\$-Funktion erstellt, genau wie die LEFT\$-Funktion, einen Teilstring von x Zeichen Länge aus dem gegebenen String. Allerdings von rechts (engl. right) beginnend.

RIGHT\$ ("SHARP",2) = "RP"
RIGHT\$ ("COMPUTER",5) = "PUTER"

MID\$ ("STRING",x,y)

Die MID\$-Funktion erstellt einen Teilstring aus dem gegebenen String. Dabei bestimmt x den Startpunkt des Teilstrings (von links beginnend), und y die Länge des Teilstrings.

MID\$ ("SHARP",3,2) = "AR"
MID\$ ("COMPUTER",3,4) = "MPU"

LEN "STRING"

Diese Funktion errechnet die Anzahl der Zeichen eines Strings. Alle Zeichen werden gezählt, auch Leerzeichen.

LEN "SHARP" = 5
LEN "COMPUTER" = 8

STR\$ x

Dies STR\$-Funktion wandelt x in einen String um.

STR\$ 12.49 = "12.49"

VAL "STRING"

Die VAL-Funktion ist die Umkehrfunktion der STR\$-Funktion. Eine in einem String gespeicherte Zahl wird in eine Zahl umgewandelt. Achtung: Wenn der String mit einem Buchstaben oder einem Zeichen beginnt, dann nimmt VAL den Wert 0 an. Die VAL-Funktion kann nur numerische Zeichen (0-9) und das E für Exponenten umwandeln.

VAL "12.49" = 12.49
VAL "A12.4" = 0
VAL "12A3" = 12

3.1.3 Logische Operatoren

AND

Der logische Operator AND verbindet Bedingungen in IF...THEN-Anweisungen.

Die AND-Bedingung ist nur wahr, wenn beide Bedingungen erfüllt sind.

OR

Der logische Operator OR verbindet Bedingungen in IF...THEN-Anweisungen.

Die OR-Bedingung ist wahr, wenn eine der Bedingungen erfüllt ist.

NOT

NOT ist ein logischer Operator, der Bedingungen in IF...THEN-Anweisungen verbindet.

Die NOT-Bedingungen ist wahr, wenn die Bedingungen in IF...THEN-Anweisungen falsch sind.

3.1.4 Programm-Anweisungen

AREAD x

Der AREAD-Befehl wird benutzt, um Programmen, die mit der DEF-Taste gestartet werden, einen Startwert zuzuweisen, der in

Do not sale !

x abgelegt wird. AREAD kann nur in der ersten Zeile nach der Zeile mit dem Label gesetzt werden. Sollte der Befehl an einer anderen Stelle im Programm auftauchen, so wird er ignoriert. Die Variable nach dem AREAD-Befehl kann sowohl eine numerische- als auch eine Stringvariable sein.

```
10 "A":AREAD A:WAIT 0
20 FOR I=1 TO 10
30 PRINT I*A
40 NEXT I
50 END
```

Wenn Sie das Programm mit DEF A starten, dann sehen Sie, daß der Wert aus dem RUN-Modus übernommen wird. Achtung: Der AREAD-Befehl hat auch seine Tücken. Sollte in der Anzeige noch ein Wert aus einem vorherigen Programm stehen, so wird auch dieser Wert übernommen. Wenn nur das Bereitschaftszeichen im Display steht, so wird bei der Ausführung des Programmes 0 bzw. ASCII-Null übernommen.

CLEAR

Mit dem CLEAR-Befehl werden alle Variablen auf Null bzw. auf leer gesetzt. Wenn CLEAR an den Anfang eines Programmes gesetzt wird, so werden Speicherbereiche, die von den Variablen anderer Programme im Rechner belegt sind, freigemacht.

```
10 D=20:DIM F(20)
20 CLEAR
30 PRINT D
40 END
```

Wirkung: Die Variable D wird auf 0 gesetzt und die den F() zugewiesenen Speicher werden freigesetzt.

CONT

Der CONT-Befehl veranlaßt den Rechner ein mit BREAK (BRK) oder STOP abgebrochenes Programm weiter abzuarbeiten. Dabei bleiben die Werte der Variablen erhalten. Auch alle anderen Zustände des Computers (DIM, GOSUB, FOR...NEXT) bleiben erhalten.

BEEP x

Mit diesem Befehl wird der Computer veranlaßt kurze Piep-Töne zu erzeugen. Leider ist es bei der Programmierung in BASIC nur beim PC 1500 möglich die Tondauer zu verändern.

BEEP 10

Achtung: Der BEEP-Befehl wird zwar nicht von allen Rechnern der PC-Reihe ausgeführt, aber er wird von allen akzeptiert.

CURSOR x

Mit dieser Anweisung wird die Cursorposition auf dem Display des Rechners festgelegt. Die CURSOR-Funktion ist nicht auf allen Rechnern enthalten. Wenn Sie in SPCS-Programmen benutzt wird, dann muß sie nur eingesetzt werden, wenn sie auch auf dem jeweiligen Rechner vorhanden ist.

CURSOR 5

Der nächste Ausdruck auf dem Display beginnt an der 5-ten Stelle des Displays.

*DIM X(Y)*8: DIM X(2,2)*

Dieser Befehl deklariert Feldvariablen (Arrays). Im Gegensatz zu einfachen Variablen müssen Feldvariablen vor ihrer ersten

Benutzung definiert werden. In der Deklaration wird die Größe des Feldes festgelegt.

```
DIM X(20)      reserviert: X(0) bis X(20)
DIM X$(20)     reserviert: X$(0) bis X$(20)
DIM X(10,10)   reserviert: X(0,0) bis X(10,10)
```

END

Dieser Befehl beendet den Programmablauf an einer beliebigen Stelle. Die END-Anweisung kann am Ende eines Programmes fortgelassen werden. Das Programm wird dann automatisch nach Ausführung der letzten Programmzeile beendet.

FOR TO STEP NEXT

Die 4 Komponenten dieses Befehls gehören fest zusammen. Sie schließen einen Programmteil ein, der in einer Schleife mehrfach durchlaufen werden soll. Läßt man den STEP-Befehl fort, dann wird die Schrittweite gleich 1 gesetzt.

```
10 FOR X=1 TO 10 STEP 1
20 PRINT X
30 NEXT X
```

GOSUB x

Durch den GOSUB-Befehl springt das Programm in ein Unterprogramm, das mit der Programmzeile x beginnt.

```
100 GOSUB 1000
110 END
1000 PRINT"UNTERPROGRAMM"
1010 RETURN
```

GOTO x

Dieser Befehl läßt den Rechner in die Programmzeile *x* des Programms springen.

```
10 GOTO 100
20 GOTO 50
50 GOTO 10
100 GOTO 20
```

GRAD

Mit dem GRAD-Befehl werden die Winkelwerte im Rechner auf Neugrad umgestellt. Der SPCS hat drei Formen der Winkeldarstellungen.

1. Alt-Grad
2. Radial-Wert
3. Neu-Grad

IF...THEN...ELSE

Durch den IF..THEN..ELSE-Befehl wird ein Befehl nur dann erfüllt, wenn eine Bedingung erfüllt ist, anderenfalls wird der Befehl nach dem ELSE ausgeführt.

```
IF A=0 THEN PRINT"HALLO" ELSE PRINT"TSCHÜB"
```

LET X=1

Durch den LET-Befehl werden Variablen Werte zugewiesen. Die LET-Anweisung kann weggelassen werden. Erfolgt die Variablenzuweisung auf eine IF...THEN...ELSE-Anweisung, dann muß der LET-Befehl stehen, da der Interpret den Variableninhalt sonst für eine Zeilennummer hält.

10 LET X=2 - Das LET kann weggelassen werden.
20 IF U=5 THEN LET B=1 - Das LET muß stehen.
30 LET A\$=A\$+"S" - Das LET kann wegfallen.

NEW

Gibt man den Befehl NEW ein, gefolgt von ENTER, so wird der gesamte Inhalt des Speichers gnadenlos gelöscht. Die Wirkung des NEW-Befehls kann von BASIC aus nicht mehr aufgehoben werden.

NEW - Der gesamte Speicherinhalt wird gelöscht.

ON X GOSUB ...

Durch den Befehl ON GOSUB springt das Programm in Abhängigkeit von X in ein Unterprogramm. Wenn der Wert von X 1 ist, springt das Programm zum ersten Unterprogramm, bei 2 zum zweiten usw. Der Wert von X muß eine positive Zahl ohne Dezimalstellen sein, die zwischen 0 und 255 liegt.

```
10 INPUT A
20 ON A GOSUB 100,200,300
30 GOTO 10
100 PRINT"100":RETURN
200 PRINT"200":RETURN
300 PRINT"300":RETURN
```

Gibt man 1 ein, so erscheint danach die Zahl 100 auf dem Display. Wenn man 2 eingibt, so erscheint eine 200 und bei 3 erscheint eine 300.

ON X GOTO ...

Der ON GOTO-Befehl arbeitet genau wie der ON GOSUB-Befehl, nur das kein Unterprogramm, sondern eine Programmzeile anvisiert wird.

REM

Mit dem REM-Befehl hat man die Möglichkeit, Bemerkungen (REMARKS) in das Programm einzufügen, ohne daß der Programmablauf beeinflußt wird. Solange genügend Speicherplatz vorhanden ist, kann man den REM-Befehl nach Belieben einsetzen. Sollte der Platz allerdings knapp werden, dann kann man hier am leichtesten Speicherplatz sparen.

```
10 FOR I=1 TO 10
20 PRINT I
30 REM DIESE ZEILE HAT KEINE WIRKUNG
40 NEXT I
```

RETURN

Wenn sich ein Programm in einem Unterprogramm befindet, so springt das Programm durch den RETURN-Befehl zu der Zeile zurück, in der sich der GOSUB-Befehl befindet.

```
10 GOSUB 100
20 PRINT "HAUPTPROGRAMM"
30 GOTO 10
100 PRINT"UNTERPROGRAMM"
110 RETURN
```

RUN

Der RUN-Befehl startet ein im Speicher befindliches Programm. Durch die zusätzliche Angabe einer Zeilennummer kann die Startzeile des Programmablaufes festgelegt werden.

Do not sale !

RUN - Startet das Programm im Speicher ab der ersten Zeilennummer.

RUN 50 - Startet das Programm ab Zeile 50.

STOP

Mit dem **STOP**-Befehl kann man den Programmablauf unterbrechen. Das Programm wird an der Stelle abgebrochen, an der der Befehl erscheint. Der Rechner quittiert den Abbruch mit "BREAK IN ...".

```
10 PRINT"HALLO"  
20 STOP  
30 PRINT"HIER KOMMT DAS PROGAMM NICHT AN"
```

TRON

Durch den **TRON**-Befehl wird die eingebaute Ablaufverfolgung eingeschaltet. Fehler, die sich in das Programm eingeschlichen haben, können so leichter gefunden werden.

TRON

TROFF

Die **TROFF**-Anweisung schaltet die oben genannte Ablaufverfolgung wieder aus.

3.1.5 Ein- und Ausgabeanweisungen

DATA x,x,x

Wenn Daten in ein Feld eingelesen werden, so läßt sich diese Aufgabe mit Hilfe des **DATA**-Befehls bewältigen. Die in der **DATA**-Zeile abgelegten Daten können mit Hilfe des **READ**-

Befehls gelesen werden. Dabei liest der erste READ-Befehl den ersten Wert der ersten DATA-Zeile, der zweite READ-Befehl den zweiten Wert usw. Sind alle Daten der Zeile ausgelesen, so wird der Lesezeiger auf die nächste DATA-Zeile gesetzt. Eine DATA-Zeile kann mit Hilfe des RESTORE-Befehls auch mehrfach gelesen werden.

```
10 DIM N(5)
20 FOR I=1 TO 5
30 READ N(I)
40 PRINT N(I)
50 NEXT I
60 END
70 DATA 10,20,30,40,50,60,70,80,90,100
```

Die Werte aus den DATA-Zeilen werden entsprechend in die Feldvariablen gelesen. $N(1)=10$, $N(2)=20$...
DATA-Zeilen haben keinen Einfluß auf den Ablauf des Programms. Sie können also an jeder beliebigen Stelle im Programm stehen.

INPUT x

Sollen Variablen während des Programmablaufes verändert werden, so eignet sich hierfür der INPUT-Befehl. Der Programmablauf wird angehalten und ein Fragezeichen erscheint im Display. Sie geben den gewünschten Wert ein, gefolgt von ENTER und dieser Wert wird der entsprechenden Variable zugeordnet.

```
10 INPUT A
20 INPUT B
30 PRINT "A * B = "; A*B
40 END
```

INPUT # X

Dieser Befehl lädt den Inhalt einer Datei von Kassette und weist die Werte zu. Die Datei muß zuvor mit dem PRINT #-Befehl

abgespeichert worden sein. Bei X kann es sich sowohl um numerische, als auch um Stringvariablen handeln, die auch dimensioniert sein dürfen.

```
10 INPUT# B
20 INPUT# B$
30 INPUT# B(2)
40 INPUT# n(*)
```

Mit INPUT# N(*) werden alle im Feld N() gespeicherten Werte geladen.

LIST

Das Kommando LIST bringt Programme, die sich im Speicher befinden, zur Anzeige. Dabei werden die Zeilennummern nacheinander von der kleinsten bis zur größten aufgelistet.

```
LIST 100 - 200
```

Die Zeilennummern von 100 bis 200 einschließlich werden aufgelistet. Wenn der Speicher durch ein Password geschützt ist, wird der LIST-Befehl nicht ausgefüllt.

PAUSE "TEXT"

Der PAUSE-Befehl veranlaßt den Computer, den angegebenen Text oder Wert zeitlich begrenzt auf dem Display auszugeben. Dabei übernimmt der PAUSE-Befehl die Funktion des PRINT-Befehls.

```
10 PAUSE "HALLO"
20 PAUSE 200
```

Auf dem Display erscheint für eine kurze Zeit das Wort HALLO, dann erscheint rechts im Display die Zahl 200

PRINT x

Der PRINT-Befehl gibt Texte oder Werte auf dem Display wieder. Die Dauer der Wiedergabe hängt vom WAIT-Befehl ab.

```
10 PRINT"SHARP PC"  
20 PRINT AS  
30 PRINT B
```

PRINT# x

Mit dem PRINT#-Befehl werden die Werte von gesetzten Variablen auf dem Kassettenrecorder abgespeichert. Mit dem PRINT#-Befehl können sowohl normale, als auch dimensionierte Variablen abgespeichert werden.

```
10 PRINT# A,B,C  
20 PRINT# A(*)  
30 PRINT AS
```

READ A,A,A

Mit Hilfe des READ-Befehls können die Daten, die in DATA-Zeilen gespeichert sind, gelesen und somit im Programm verwendet werden. Der RESTORE-Befehl setzt den Zeiger für die zu lesenden Daten wieder auf den ersten DATA-Wert.

```
10 FOR I=1 TO 10  
20 READ AS  
30 PRINT AS;  
40 NEXT I  
50 DATA "S","H","A","R","P","P","C"  
60 RESTORE  
70 GOTO 10
```

RESTORE

Setzt den Zeiger der zu lesenden Daten auf die erste DATA-Zeile im Programm. Daten, die schon ausgelesen worden sind, müssen durch den RESTORE-Befehl wieder "aktiviert" werden.

```
10 FOR I=1 TO 10
20 READ D
30 PRINT D*I
40 NEXT I
50 RESTORE
60 GOTO 10
70 DATA 1,2,3,4,5,6,7,8,9
```

USING "##.##"

Der USING-Befehl formatiert die Ausgabe auf dem Display. Die mit USING erstellte Formatierung hat für alle folgenden Ausgaben Gültigkeit. Die Art der Formatierung wird durch die folgenden Spezifikationen festgelegt:

1. # - Rechtsbündiges Feld für numerische Zeichen
2. . - Der zu setzende Dezimalpunkt
3. ^ - Zahlen in Exponentialschreibweise
4. & - Linksbündiges Feld für alphanumerische Zeichen

WAIT 150

Mit dem WAIT-Befehl wird die Länge der Ausgabezeit auf dem Display festgelegt. Jeder Zähler im Argument des Befehls erhöht die Dauer der Ausgabezeit um eine 1/64 Sekunde. Also legt WAIT 64 die Ausgabe auf etwa 1 Sekunde fest.

10 WAIT 64
20 PRINT"1 SEKUNDE"
30 WAIT 128
40 PRINT"2 SEKUNDEN"

3.1.6 Programmspeicherung

CLOAD

Mit Hilfe des CLOAD-Befehls werden Programme vom Kassettenrecorder in den Speicher geladen. Dieser Befehl kann logischer Weise nur in Verbindung mit einem Kassettenrecorder benutzt werden.

CLOAD - lädt das nächste Programm von Kassette ein.
CLOAD"TEST" - lädt das Programm TEST von Kassette ein.

CLOAD?

Der CLOAD?-Befehl vergleicht das Programm im Speicher mit einem Programm auf Kassette. Der Befehl dient zur Überprüfung von abgespeicherten Programmen. Natürlich kann auch dieser Befehl nur in Verbindung mit einem Kassettenrecorder genutzt werden.

CLOAD? - vergleicht das Programm im Speicher mit dem nächsten von Kassette.

CLOAD?"TEST" - vergleicht das Programm im Speicher mit dem Programm TEST auf Kassette.

CSAVE

Mit dem CSAVE-Befehl werden Programme, die sich im Speicher befinden, auf Kassette gespeichert. Dieser Befehl kann, wie

alle Programmspeicherungsbefehle, nur in Verbindung mit einem Kassettenrecorder benutzt werden.

CSAVE - speichert Programm ohne Bezeichnung auf Kassette.

CSAVE"TEST" - speichert Programm mit Namen TEST auf Kassette.

CSAVE"TEST","OHO" - speichert Programm mit Namen TEST und Passwort OHO auf Kassette.

3.1.7 Druckerausgabe

LLIST

Der **LLIST**-Befehl druckt das Programm auf dem angeschlossenen Drucker oder Plotter aus. Voraussetzung dafür ist natürlich, daß ein Drucker an den Rechner angeschlossen ist.

LLIST Gibt das ganze Programm auf dem Drucker aus.

LLIST 10 Druckt nur die Zeile 10 des Programms aus.

LLIST 10-60 Gibt die Zeilen 10 bis 60 auf dem Drucker aus.

LLIST -100 Druckt die Programmzeilen bis 100 aus.

LLIST 100- Gibt die Programmzeilen ab 100 bis zum Ende aus.

LPRINT"SHARP"

Mit dem **LPRINT**-Befehl werden Texte oder Werte auf dem Drucker ausgegeben. Ein Drucker ist auch hierfür die notwendige Voraussetzung.

Do not sale !

LPRINT"HALLO" - gibt HALLO auf dem Drucker aus.
LPRINT A\$ - druckt den Inhalt des Strings A\$ aus.

Do not sale !

Do not sale !

Anhang A

Alphabetische Programmliste

Abschreibungen.....	45
Brüche.....	21
Calc.....	47
Darlehen.....	86
Frosch.....	144
ggT und kgV.....	28
High Noon.....	137
Henker.....	121
Jackpot.....	124
Kegel.....	90
Kim.....	131
Körper.....	26
Lotto 1 und 2.....	82, 83
Monitor.....	105
Optimal.....	58
Orakel.....	96
Phytagoras.....	34
Primzahl.....	117
ReLi.....	89
Rekneh.....	129
Restwert.....	78
RND - RND.....	109
RND-Test.....	111
Schloß.....	126
Shuttle.....	112
Spezroule.....	93
Sprit.....	74
Systeme.....	114
Tage.....	98
Takt.....	75

Torten	139
Turf	134
U-Boot.....	141
Weltweit	107
Zinseszins.....	84

Do not sale !

Anhang B

ASCII-Tabelle

CODE	CHAR.	CODE	CHAR.	CODE	CHAR.
32	SPACE	33	!	34	"
35	#	36	\$	37	%
38	+	39	'	40	(
41)	42	*	43	+
44	,	45	-	46	.
47	/	48	0	49	1
50	2	51	3	52	4
53	5	54	6	55	7
56	8	57	9	58	:
59	;	60	<	61	=
62	>	63	?	64	a
65	A	66	B	67	C
68	D	69	E	70	F
71	G	72	H	73	I
74	J	75	K	76	L
77	M	78	N	79	O
80	P	81	Q	82	R
83	S	84	T	85	U
86	V	87	W	88	X
89	Y	90	Z		

Do not sale !

Do not sale !

Anhang C

Alphabetisches Verzeichnis der BASIC-Befehle

Numerische Funktionen:

ABS	149
ACS	149
ASN	149
ATN	150
COS	150
DEG	150
DMS	151
EXP	151
INT	151
LN	151
LOG	152
PI	152
RND	152
SGN	152
SIN	153
SQR	153
TAN	153

Logische Operatoren:

AND	157
NOT	157
OR	157

String-Funktionen:

ASC	154
CHR\$	154
INKEY\$	154
LEFT\$	155
LEN	156
MID\$	156
RIGHT\$	155
STR\$	156
VAL	156

Programm-Anweisungen:

AREAD	157
BEEP	159
CLEAR	158
CONT	159
CURSOR	159
DIM	159
END	160
FOR TO NEXT STEP NEXT	160
GOSUB	160
GOTO	161
GRAD	161
IF THEN	161
LET	161
NEW	162
ON GOSUB	162
ON GOTO	163
REM	163
RETURN	163
RUN	163
STOP	164
TRON	164
TROFF	164

Ein- und Ausgabeanweisungen:

DATA	164
INPUT	165
INPUT#	165
LIST	166
PAUSE	166
PRINT	167
PRINT#	167
READ	167
RESTORE	168
USING	168
WAIT	168

Programmspeicherung:

CLOAD	169
CLOAD?	169
CSAVE	169

Druckerausgabe:

LLIST	170
LPRINT	170

Germann
Jörg
Waldvogel

**Das große
Pocket
Computer
Buch**

SHARP

Ein DATA BECKER Buch

Do not sale !

Lernen Sie Ihren SHARP-Pocket-Computer richtig kennen. Dieses Buch erklärt Ihnen alles, was Sie schon immer über Ihren SHARP-Pocket-Computer wissen wollten. Von einfachen BASIC-Programmen über komplexe Maschinenroutinen bis hin zu Hardwarekniffen – da bleibt kein Wunsch offen.

Aus dem Inhalt:

- Aufbau des Speichers
- Verwaltung von Variablen
- Interessante Systemadressen
- Der Zeichengenerator
- Programmertips für BASIC-Programme
- Knacken des PASS-Worts
- Die CPU: Das Herz des Rechners
- Der Assembler: Installation und Bedienung
- Befehle zur Steuerung der Ports
- Grafik: Turbo Graphics
- Mit Draht und Lötzinn

Germann, Jörg, Waldvogel
Das große Sharp-Pocket-Computer-Buch
416 Seiten, DM 39,-
ISBN 3-89011-177-7

Do not sale !