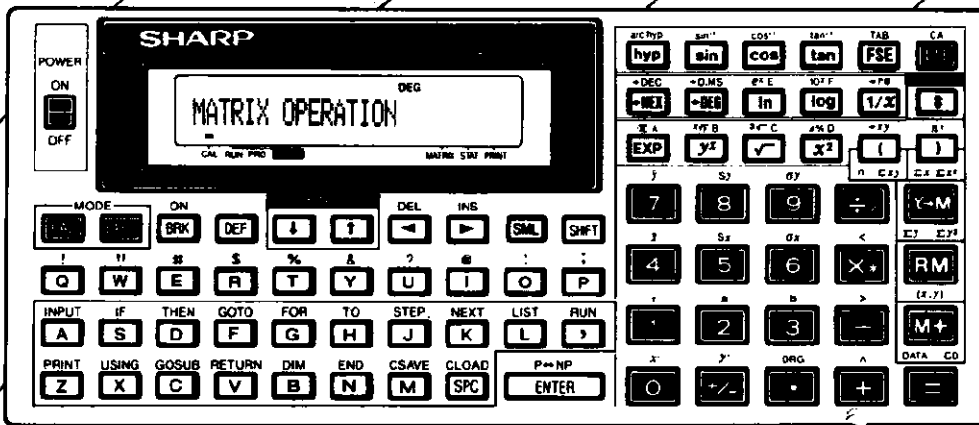


Der SHARP in Deiner Hand



**Die Programm-Sammlung
für alle
SHARP Pocket-Computer.**

1.	VORWORT.....	5
2.	ZUBEHÖR FÜR SHARP-POCKETCOMPUTER.....	7
3.	TIPS UND TRICKS.....	13
4.1	MATHEMATIK.....	26
1.	Kurvendiskussion.....	27
2.	Polynomdivision I.....	31
3.	Polynomdivision II.....	32
4.	Polynommultiplikation.....	34
5.	Interpolation nach Lagrange.....	35
6.	Gleichungen 3. Grades.....	37
7.	Lineares Gleichungssystem.....	38
8.	Relativitätstheorie.....	39
9.	Statistische Berechnungen.....	40
10.	Bruchapproximation.....	43
11.	Gleichungen 2., 3. und 4. Grades.....	43
12.	Matrizenmultiplikation.....	45
13.	Matrizeninversion.....	46
14.	Matrizenorthogonalisierung.....	47
15.	Integral.....	48
16.	Gauss'scher Algorithmus.....	49
17.	Nullstellen Berechnung.....	50
18.	Differentiation.....	50
19.	Cholesky.....	51
20.	Quadratische Gleichung.....	53
21.	Größter gemeinsamer Teiler.....	53
22.	Kleinstes gemeinsames Vielfaches.....	54
23.	Regula Falsi (Nullstellen).....	54
24.	Vektorberechnungen.....	55
25.	Rechnen mit komplexen Zahlen und Gleichungssystemen.....	59
26.	Finanzmathematik.....	63
27.	Einlagenrechnung.....	67
28.	Rentenrechnung.....	69
29.	Dreiecksberechnung.....	72
30.	Kreisabschnitt.....	75
31.	Bruchrechnen.....	76
32.	Arithmetische Folgen und Reihen.....	78
33.	Geometrische Folgen und Reihen.....	80
34.	Schiefe Würfe.....	81
4.2	ELEKTRONIK UND EDV.....	84
1.	Dezimal-Binär-Umwandlung.....	85
2.	Dezimal-Hexadezimal-Binär-Umwandlung.....	85
3.	Hamming-Code.....	90
4.	Wheatstone-Brücke.....	92
5.	Einfaches Schaltinterface.....	94
6.	Oszillator NE 555.....	96
7.	Zenerdiode.....	97
8.	Stern-Dreieck.....	99
9.	Kondensatoren-Berechnung.....	100
10.	Belasteter Spannungsteiler.....	102
11.	Serien-Parallel-Umwandlung.....	102
12.	Schwingkreis.....	103
13.	Vitrometer.....	104

4.3	MASCHINENBAU.....	107
	1. Festigkeitsberechnung.....	108
	2. Momentensatz.....	108
	3. Riemengetriebe.....	110
	4. Brinellhärte.....	111
	5. Statik.....	111
	6. Sparrenberechnung.....	115
4.4	SPIEL & SPAB.....	117
	1. Reaktionstest.....	118
	2. Frogger.....	118
	3. Buchstaben erraten.....	121
	4. Ghostbuster.....	122
	5. Siebzehn und Vier.....	124
	6. Mastermind.....	126
	7. Denktest.....	127
	8. 51 gleich tot.....	129
	9. Barrikade.....	131
	10. Mondlandung.....	133
	11. Hangman.....	135
	12. Senso.....	137
	13. Shipdown.....	138
	14. Airplane.....	142
4.5	SONSTIGE FACHGEBIETE.....	145
	1. Biorythmus.....	146
	2. Körperoberfläche.....	148
	3. Stundenplan.....	149
	4. Großschrift mit CE 126.....	151
	5. Namenserkennung.....	153
	6. Mini-Menü.....	154
	7. Telefongebührenzähler.....	155
	8. Digitaluhr.....	156
	9. Laufzeitermittlung.....	156
	10. Kalender.....	158
	11. Canasta-Abrechnung.....	160
	12. Ilford-Verfahren.....	161
	13. Ilford-Cibachrome-Verfahren.....	162
	14. Tentel-UK3-Prozeß.....	163
	15. Einheitenumrechnung.....	164
	16. Chemische Formeln.....	166
	17. Gehäuseberechnung von Lautsprechern.....	169
	18. Berechnung von aktiven Filtern.....	174
	19. Reisekonto.....	185
	20. Seenavigation.....	190
	21. Berechnung der Molmasse.....	196
	22. Adress-Datei.....	198
	23. Flugvorbereitung.....	203
	ANHANG A.....	212
	ANHANG B.....	214

Kapitel 1

Vorwort

Vorwort

Wir beglückwünschen Sie zum Kauf von 'Der SHARP in Deiner Hand'.

Wie ist diese Programmsammlung entstanden ?

TRANSFILE ist die Standard-Rechnerkopplung mit der Sie Ihren SHARP PC an einen Home- bzw. Personalcomputer anschließen können, um Ihre SHARP-Programme auf den Disketten des 'großen Bruders' abspeichern zu können.

Von den TRANSFILE-Benutzern wurden uns zahlreiche professionelle Programme auf Diskette zugesendet, die als Public Domain Programme auf 3 Disketten vertrieben werden. Von diesen Public Domain Programmen sind die besten Programme in diesem Buch zusammengefaßt.

Dadurch sind die Programme auch allen SHARP-Benutzern zugänglich, die keinen 'großen' Computer besitzen und somit die Programme auf den Disketten nicht benutzen können.

Alle Programme wurden direkt so übernommen, wie Sie uns von den Programmierern eingesendet worden sind. Die Dokumentation wurde überarbeitet und soweit es notwendig bzw. möglich war noch entsprechend ergänzt.

Die Programme wurden, soweit es möglich war, überprüft. Nur da viele Autoren dazu beigetragen haben dieses Werk zu erstellen, kann die Programmsammlung so preiswert angeboten werden. Die Autoren werden namentlich in diesem Buch erwähnt.

Hiermit möchten wir nochmals allen Autoren recht herzlich danken, die an diesem Werk beteiligt waren.

Und nun wünsche ich Ihnen viel Spaß !

Peter Dankel

Hinweis:

Alle Rechte vorbehalten !

Die Programmsammlung und die Programme sind urheberrechtlich geschützt. Diese dürfen nur mit schriftlicher Genehmigung von Yellow Computing vervielfältigt, in andere Sprachen übersetzt oder in Datenverarbeitungsanlagen erfaßt werden.

Trotz gewissenhafter Prüfung kann für Fehler in der Anleitung oder an Programmen keine Haftung übernommen werden.

Kapitel 2

Zubehör für SHARP-Pocketcomputer

Zubehör für Sharp-Pocketcomputer

Zusätzlich zu den von der Firma SHARP angebotenen Zusatzgeräten zu den Pocketcomputern, gibt es mittlerweile sehr viel Zubehör von verschiedenen anderen Anbietern, welches wir hier in einer kurzen Übersicht vorstellen wollen. Die Verkaufspreise sind die Preise die wir bei der Erstellung der Programmsammlung in Erfahrung bringen konnten. Für die Vollständigkeit und Richtigkeit der Angaben übernehmen wir keine Haftung. Sie können diese Produkte bei Ihrem Fachhändler erhalten.

Kassetteninterface (für SHARPs mit 11pol. Anschluß) 29.- DM

Mit dem Kassetteninterface können die Programme und Daten der SHARP-Pocketcomputer auf Kassette gespeichert werden. Das Kassetteninterface kann an jeden handelsüblichen Kassettenrekorder mit 3.5mm Klinkenbuchsen angeschlossen werden.

Kommunikationsinterface (für SHARPs mit 11pol. Anschluß) 29.- DM

Mit dem Kommunikationsinterface können die Programme und Daten direkt zwischen zwei SHARP-Pocketcomputern ausgetauscht werden.

Poweradapter (für SHARP PCs mit 11pol. Anschluß) 39.- DM

Um die Batterien der Pocketcomputer zu schonen gibt es den Poweradapter. Der Poweradapter wird an die 11pol. Schnittstelle angeschlossen und versorgt den Rechner mit Strom. Es wird automatisch zwischen externer und interner Stromversorgung umge schaltet.

Batteriebox (für SHARP PCs mit 11pol. Anschluß) 29.- DM

Um die Batterien der Pocketcomputer zu schonen gibt es die Batteriebox. Die Batteriebox wird an die 11pol. Schnittstelle angeschlossen und versorgt den Rechner mit Strom. Es wird automatisch zwischen externer und interner Stromversorgung umge schaltet.

Speichererweiterungen auf Anfrage

Bei vielen SHARP-Pocketcomputern kann der RAM-Speicher durch Auf-rüstung erweitert werden. Da die Preise der RAM-Bausteine sehr stark schwanken und da es sehr viele verschiedene Größen von Speichererweiterungen gibt, kann hier leider keine genaue Aussage über die Preise gemacht werden.

Centronics-Interface

199.- DM

Mit dem Centronics-Interface können die SHARP-Pocketcomputer an einen Matrixdrucker mit Centronics-Schnittstelle angeschlossen werden. Dies erlaubt den Ausdruck von 80 Zeichen breiten Listings mit den Pocketcomputern.

TRANSFILE

TRANSFILE koppelt die unterschiedlichsten SHARP-Pocketcomputer mit den verschiedenen Home- und Personalcomputern, wie ATARI ST, PC/XT/AT, AMIGA, C64 und APPLE II.

Mit TRANSFILE können die Programme und Daten der 'kleinen' Pocketcomputer auf die 'großen Brüder' übertragen werden. Dadurch ist es möglich, die Programme der 'kleinen' Rechner auf den Massenspeichern (Diskette, Festplatte) der 'großen' Rechner abzuspeichern. Die Programme der Pocketcomputer können auf dem Personalcomputer mit Hilfe eines Editors erstellt oder verändert werden. Damit können Programme für die Pocketcomputer sehr übersichtlich, schnell und professionell mit einer guten Tastatur und normalem Monitor entwickelt werden.

Natürlich können die Daten und Programme auch wieder auf den Pocketcomputer zurück übertragen werden.

Weiterhin können die Daten und Programme auf einem Matrixdrucker ausgedruckt werden, was besonders bei der Programmdokumentation bzw. bei der Fehlersuche sehr von Vorteil ist.

Das TRANSFILE-Paket besteht jeweils aus Diskette, Handbuch und dem entsprechenden Interface.

Im Anschluß werden die Vorteile der einzelnen TRANSFILE-Programme für die unterschiedlichen Home- bzw. Personalcomputer aufgezeigt.

TRANSFILE ST PLUS

129.- DM

TRANSFILE ST PLUS für den anspruchsvollen Anwender. Mit MERGE, RE-NUMBER, Disassembler und Schnittstellen zu OMIKRON-BASIC und zu einem beliebigen Editor. TRANSFILE ST PLUS erlaubt die Rechnerkopplung zwischen den SHARP-Pocketcomputern PC 1100/50, PC 1245/46/47/48/51/60/61/62/80, PC 1401/02/03/21/25/30/45/50/60/75 und PC 1350/60 und allen ATARI ST Computern, auch den MEGA ST Modellen.

TRANSFILE ST 1500/2500

149.- DM

Diese TRANSFILE ST Version entspricht im wesentlichen TRANSFILE ST PLUS und ist für SHARP PC 1500 und für SHARP PC 2500 geeignet. Zum Anschluß an den PC 1500 wird das CE-150 benötigt.

TRANSFILE ST 1600

129.- DM

Version von TRANSFILE ST für den SHARP PC 1600. Läuft ebenfalls auf allen ATARI ST Computern sowie den MEGA ST Modellen. Unterstützt den PC 1600 auch im PC 1500-Modus. Mit Editor-Schnittstelle.

TRANSFILE ST TOOLBOX

79.- DM

Die Toolbox wurde für alle TRANSFILE-Anwender entwickelt, die auf der einen Seite fertige Programme wünschen, um Ihre mobil erfassten Daten in ADIMENS oder 1ST-WORD-Format zu konvertieren, die aber auf der anderen Seite gerne eigene, kleine Programme in OMIKRON.BASIC zu TRANSFILE ST entwickeln möchten.

Deshalb sind in der Toolbox fertige Programme integriert, die ein abgeschlossenes, sofort verwendbares Programm darstellen aber gleichzeitig vom Anwender auf seine eigenen Bedürfnisse angepasst oder erweitert werden können.

Alle Programme sind vollständig in OMIKRON.BASIC geschrieben, so daß auch weniger geübte Programmierer die Programme erweitern können.

TRANSFILE ST ASSEMBLER

99.- DM

Hierbei handelt es sich um einen makrofähigen, symbolischen Crossassembler für die SHARP CPU SC 61860. Damit ist es endlich möglich, komfortabel Assemblerprogramme für die SHARP Pockets zu erstellen. Weiterhin enthält das Programmpaket einen symbolischen Disassembler. Beide Programmteile arbeiten direkt mit TRANSFILE ST PLUS zusammen.

TRANSFILE PC PLUS (Version 5.0)

129.- DM

TRANSFILE PC PLUS erlaubt die Kopplung zwischen den SHARP-Pocketcomputern PC 1245/46/47/48/51/60/61/62/80, PC 1401/02/03/21/25/30/45/50/60/75 und PC 1350/60 und allen PC/XT/AT und kompatiblen Rechnern.

Neue, sehr stark erweiterte TRANSFILE-Version mit komfortablen, integriertem Full-Screen-Editor.

TRANSFILE PC 1500/2500 (Version 5.0)

179.- DM

Entspricht TRANSFILE PC PLUS, ist aber für SHARP-Pocketcomputer PC 1500 und 2500 geeignet. Mit integriertem Full-Screen-Editor.

TRANSFILE PC 1600

129.- DM

Rechnerkopplung zwischen dem SHARP-Pocketcomputer PC 1600 und allen PC/XT/AT und kompatiblen Rechnern. Mit integriertem Full-Screen-Editor.

TRANSFILE AMIGA PLUS

129.- DM

TRANSFILE AMIGA PLUS erlaubt die Rechnerkopplung zwischen SHARP-Pocketcomputern PC 1245/46/47/48/51/1260/61/62/80, PC 1401/02/03/21/25/30/45/50/60/75 und PC 1350/60 und allen AMIGA Rechnern, also AMIGA 500, 1000, 2000. Mit integrierter Editor schnittstelle.

TRANSFILE 64

99.- DM

Rechnerkopplung zwischen SHARP-Pocketcomputern PC 1260/61/62, PC 1401/02/03/21/25/30/45/50/60 und PC 1350 und allen Commodore C64, C128, SX64 Computern.

TRANSFILE II

99.- DM

Rechnerkopplung zwischen SHARP-Pocketcomputern PC 1260/61/62, PC 1401/02/30/50 und PC 1350 und folgenden APPLE II Rechnern: APPLE II, APPLE II+, APPLE IIe, aber nicht APPLE IIc !!

TRANSFILE CPC

Rechnerkopplung zwischen SHARP-Pocketcomputern PC 1260/61/62, PC 1401/02/03/21/30/50 und PC 1350 und Schneider CPC Modellen wie CPC 464, CPC 664 und CPC 6128 Computern.

PD-DISKETTEN

10.- DM

Die PD-Disketten enthalten SHARP-Basic-Programme im TRANSFILE-Format, die von TRANSFILE-Usern entwickelt und dem PD-Pool zur Verfügung gestellt wurden. Die PD-Disketten dürfen kostenlos weiterkopiert werden. Zur Zeit sind je Computer-Typ vier verschiedene Disketten verfügbar. Die PD-Software ist für TRANSFILE ST (3 1/2"), TRANSFILE AMIGA (3 1/2") und TRANSFILE IBM (3 1/2" und 5 1/4") verfügbar.

YELLOW DRIVE PC

299.- DM

YELLOW DRIVE PC ist eine vollständige Emulation einer SHARP-Floppy vom Typ CE-140F auf einem PC/XT/AT oder kompatiblen Computer. Sie können damit die Diskettenbefehle der neueren SHARP-PCs nutzen, um Ihre Programme und Daten schnell und sicher auf der Festplatte oder Diskette des PC/XT/AT abzuspeichern.

Die Übertragung mit LOAD bzw. SAVE ist bis zu 14mal schneller als mit den CLOAD bzw. CSAVE-Befehlen. Sie brauchen kein zusätzliches Programm im SHARP-PC, da dieser die notwendigen Befehle bereits eingebaut hat.

Das Programmpaket ist komplett mit Interface, Diskette und Handbuch und ist für PC 1280, PC 1360 und PC 1403/25/60/75 geeignet.

Diskette zum Buch (für ATARI ST, PC/XT/AT, AMIGA)

10.- DM

Um nicht alle Programme der Programmsammlung "Der SHARP in Deiner Hand" selbst eintippen zu müssen, gibt für die TRANSFILE-Besitzer die Diskette zum Buch.

Kassette zum Buch

20.- DM

Um nicht alle Programme der Programmsammlung "Der SHARP in Deiner Hand" selbst eintippen zu müssen, gibt für die SHARP-Besitzer die Kassette zum Buch.

Bücher zu SHARP Pocketcomputern

Es gibt sehr viele Bücher für SHARP-Pocketcomputer die aber hier aus Platzgründen nicht alle aufgezählt werden können. Selbst wenn hier jetzt 3 Seiten Bücher aufgezählt würden, so würde diese Aufzählung nichts über den Inhalt aussagen. Sie können sich eine Bücherliste bei Ihrem Fachhändler oder in einer Buchhandlung geben lassen bzw. die Bücher dort direkt anschauen. Es soll hier nur ein kurzer Überblick über die verschiedenen Arten der Bücher gegeben werden.

Die Bücher sind meist für einen oder eine kleine Gruppe von SHARP-Rechnern geeignet, wobei Sie unbedingt darauf achten sollten, daß Ihr Rechnertyp genannt wird.

Systemhandbuch: Beschreibt den internen Aufbau des jeweiligen Pocketcomputers (Speicherlage, spezielle Adressen, Tricks)

Maschinensprachebuch: Beschreibt, wie der Pocketcomputer in Maschinensprache zu programmieren ist. Das Systemhandbuch ist dazu eine sehr sinnvolle Ergänzung.

Anwendungshandbuch: Enthält teilweise fertige Anwendungen und zeigt den Umgang bzw. Anwendungen mit dem Pocketcomputer.

Hardwareentwicklung: Zeigt dem Anwender, wie Hardware an den Pocketcomputer angeschlossen werden kann.

Programmsammlung: Es gibt verschiedene Programmsammlungen für Pocketcomputer, wobei diese teilweise nach den Fachgebieten geordnet sind (kaufmännisch, technisch, betriebswirtschaftlich, Spiele, Mathematik, Physik). Die Programmsammlungen enthalten fertige Programme.

Tips und Tricks: In diesen Büchern werden Tips und Tricks beim programmieren bzw. beim Umgang mit dem Pocketcomputer aufgezeigt.

Um nicht das falsche Buch zu kaufen empfehlen wir Ihnen, das Buch vorher anzuschauen. Sie können die Bücher auch nur zur Ansicht in einer Buchhandlung bestellen.

Kapitel 3

Tips und Tricks

Programmieren in Sharp-Basic

Diese kleine Einführung ist für alle gedacht, die keine oder nur wenig Erfahrung mit programmierbaren Basicrechnern haben. Sie ist als eine Art Kochrezept gedacht, mit der anhand von einigen wenigen Schritten einfache Probleme schnell und effizient gelöst werden können, ohne daß umständlich im Bedienungshandbuch gesucht werden muß.

Programmieren von Funktionen:

Besonders beim Zeichnen von Graphen der unterschiedlichsten Funktionen ist es lästig, wenn man die ganze Gleichung andauernd wieder eintippen muß, um die notwendigen Y-Werte zu berechnen. Hier kann durch programmieren der Funktion eine wesentliche Erleichterung geschaffen werden.

Beispiel:

Gegeben ist die Funktion: $y=2*x^3+3*x-2.5$ d.h. $y=2x^3+3x-2,5$

Um diese Funktion zu programmieren, benötigen wir folgende Zeilen:

```
10 INPUT "X=";X
20 Y=2*X^3+3*X-2.5
30 PRINT "Y=";Y
40 GOTO 10
```

Erklärung des Programms:

Der Befehl INPUT "X=";X zeigt zunächst X=— auf der Anzeige an, und erwartet dann die Eingabe des X-Wertes. Würde man das "X="; weglassen, so zeigt der Rechner nichts an, sondern erwartet sofort die Eingabe des X-Wertes. Es ist aber nicht sinnvoll, auf diesen Teil zu verzichten, da man ziemlich schnell vergißt, welche Eingabe der Rechner verlangt und es so zu unnötigen Fehlern kommt.

In Zeile 20 ist die Funktion programmiert. Dies geschieht in der üblichen einzeiligen Schreibweise. WICHTIG: Gegebenenfalls Klammern setzen! Hier werden die meisten Fehler gemacht. Der Befehl PRINT "Y=";Y ist ein Anzeigebefehl, der zuerst Y= anzeigt, direkt vom Ergebnis der Rechnung gefolgt. Wichtig: Strichpunkt nicht vergessen, sonst wird das Ergebnis nicht richtig angezeigt.

In Zeile 40 springt das Programm wieder an den Anfang, und es kann sofort der nächste X-Wert errechnet werden. Wird Zeile 40 weggelassen, so muß nach jeder Rechnung das Programm neu gestartet werden, was aber eigentlich Unsinn ist, da so die Programmierung keinen großen Vorteil mit sich bringen würde. Das Ziel der Programmierung ist es ja, mit wenigen Tasten drücken rechnen zu können und so an Zeit zu sparen. Beendet wird das Programm dann mit der BRK-Taste (Unterbrechungstaste)

Eingabe des Programms:

Nach dem Einschalten des Rechners ist er im CAL-Modus, durch zweimaliges Drücken der BASIC-Taste wird er in den PRO-Modus gebracht. Jetzt kann mit der Eingabe des Programmes begonnen werden.

Zuerst geben wir die Zeilennummer ein (rechte Zahlentastatur). Nun folgt der eigentliche Befehl. Dieser kann auf drei verschiedene Weisen eingegeben werden:

- 1- Man tippt den Befehl buchstabenweise ein
- 2- Man drückt die orangene SHIFT-Taste, gefolgt von der Taste, über der mit roter Schrift der entsprechende Befehl steht. Z.B. für INPUT die Taste 'A', wobei es allerdings nicht für alle Befehle eine solche Kurztaste gibt.
- 3- Man tippt die Befehle in abgekürzter Form ein. Hinweise hierzu siehe im Artikel "BEFEHLSABKÜRZUNGEN" in dieser Programmsammlung

Dann wird der Rest der Zeile eingegeben. Alle roten Zeichen und Funktionen sind nur über die SHIFT-Taste zu erreichen ! Am Ende jeder Zeile muß die ENTER-Taste gedrückt werden, um dem Rechner mitzuteilen, daß die Eingabe dieser Zeile fertig ist.

Ist das ganze Programm eingetippt, so kann es mit dem Befehl LIST angeschaut und auf Tippfehler überprüft werden. Nach der Eingabe

SHIFT L ENTER (LIST-Befehl)

wird die oberste Zeile des Programms angezeigt. Durch das Drücken der v-Taste kann nun die nächste Programmzeile betrachtet werden. Ist eine Zeile länger als die angezeigten 16 Stellen, so kann die Anzeige mit der > Taste nach rechts bzw. mit der < Taste nach links verschoben werden. Mit der ^-Taste kann man wieder eine Zeile höher gehen.

Starten des Programms:

Um das Programm nun zu starten, muß wieder die BASIC-Taste gedrückt werden. Der Rechner befindet sich jetzt im RUN-Modus. WICHTIG: Programme können nur im RUN-Modus gestartet werden, sonst erfolgt die Meldung ERROR 9 (d.h. allgemeiner Bedienungsfehler). Im RUN-Modus können aber auch keine Programme gelistet oder eingegeben werden, dies ist nur im PRO-Modus möglich.

Wir geben jetzt ein:

SHIFT , ENTER (RUN-Befehl)

Damit ist das Programm gestartet. Auf der Anzeige erscheint X=_ und wir können nun den entsprechenden Wert gefolgt von ENTER eingeben.

Danach erscheint Y= gefolgt vom Ergebnis der Rechnung. Wird jetzt wieder ENTER gedrückt, so erscheint nochmals das X= und wir können eine neue Rechnung durchführen.

Sollte es vorkommen, daß zu früh auf ENTER gedrückt wurde, z.B. bevor wir uns das Ergebnis merken konnten, so muß die Rechnung nicht nochmals durchgeführt werden. Es genügt hier, einmal auf ENTER zu drücken, ohne das vorher etwas eingegeben wurde, und schon wird der alte Wert wieder angezeigt.

Wollen wir das Programm beenden, so drücken wir entweder die BRK-Taste, CAL-Taste oder die BASIC-Taste. Wird die BASIC-Taste gedrückt, so wird gleichzeitig auch in den PRO-Modus umgeschaltet. Drücken wir die CAL-Taste so sind wir wieder im Taschenrechnermodus.

Programmieren einer neuen Funktion:

Wenn wir nun die alte Funktion nicht mehr benötigen, so muß eine neue Funktion programmiert werden. Dazu müssen wir aber nicht das ganze Programm neu eingeben. Es genügt in diesem Falle nur die Zeile 20 neu zu programmieren, da an der Eingabe bzw. der Anzeige nichts geändert werden soll.

Dazu bringen wir den Rechner in den PRO-Modus. Dies geschieht durch das Drücken der BASIC-Taste. Dadurch wird sowohl das laufende Programm beendet als auch gleichzeitig der Modus umgeschaltet.

Nun tippen wir die Zeilennummer 20 gefolgt von Y= und der neuen Funktion ein. Dann wird wieder in den RUN-Modus geschaltet, und das Programm gestartet. Jetzt kann wieder wie oben beschrieben mit der neuen Funktion gerechnet werden.

Das Umschreiben von Funktionen in Basic:

Das größte Problem bei dieser Art von Programmen bereitet dem Anfänger meist das Umschreiben von Formeln in die beim Basic notwendige einzeilige Schreibweise. Wenn man dabei jedoch einige Punkte beachtet, so ist dies leicht zu verstehen.

Beispiel:

Wir haben die Formel $Y = \sin X + \sqrt{\frac{\tan^2 x}{2x}}$

Um diese Formel dem Rechner verständlich zu machen, müssen wir diese in einzelne Ebenen aufteilen.

Die erste Ebene ist der Ausdruck 'sin X', die zweite Ebene ist die Wurzel. Die Wurzel selbst wird dann wieder in zwei Ebenen zerlegt, nämlich Nenner und Zähler.

Dazu müssen wir die "Rangordnung" der einzelnen Operationen kennen. In der Programmiersprache BASIC gilt folgende Rechenhierarchie:

- 1 - Klammern
- 2 - Abruf von PI, E, und Variablen
- 3 - Exponentiation
- 4 - Vorzeichen + -
- 5 - Multiplikation, Division
- 6 - Addition, Subtraktion
- 7 - Vergleichsoperationen < >
- 8 - Logische Operationen (und/oder/nicht)

(Operation 1 hat den höchsten und 8 den niedrigsten Vorrang)

Weiterhin sind die Operationszeichen für Multiplikation = * und für Division = /. Die Potenzen werden mit ^ dargestellt (z.B.: ^3 = x).

Einige wichtige Ausdrücke in BASIC lauten:

SQR = Wurzel
 SQU = Quadratfunktion
 RCP = Reziprokwert
 PI = Zahl Pi
 LOG = 10er Logarithmus
 LN = e-Logarithmus
 EXP = Exponentialfunktion

u. s. w.

Für nähere Erklärungen sei auf das Handbuch Seite 212 (Erklärung des mathematischen BASIC Befehlsvorrates) verwiesen. Hier werden alle BASIC-Befehle in Kurzform erklärt. Als wichtigen Punkt sollte man sich vor allem merken, daß in BASIC immer zuerst die Funktion und dann die Variable eingegeben wird. Also z.B. RCP 5 aber nicht 5 RCP !!

Um obige Funktion nun einzugeben, zerlegen wir sie in drei Blöcke:

- 1.) $\sin x$
- 2.) $(\tan x)^2$
- 3.) $(2 * x)$

Nun setzen wir die drei Blöcke zusammen und ergänzen die Wurzel. Das Ergebnis lautet: $Y = \sin X + \text{SQR}((\text{TAN } X)^2 / (2 * X))$

Wichtig ist dabei das richtige Setzen der Klammern ! Die meisten Fehler werden durch falsch gesetzte oder vergessene Klammern verursacht. Die Funktion kann anhand der bekannten Rechenregeln (Punkt vor Strich usw.) leicht auf ihre Richtigkeit überprüft werden.

Ergänzen des Programms:

Nehmen wir einmal an, wir müßten die Wertetabelle einer Funktion erstellen, und zwar für alle X-Werte von -5 bis 5 in 0.5er Schritten. Dann müßten wir 20 mal den X-Wert eingeben und vom Programm rechnen lassen. Dies kann aber bequemer durch eine Programmergänzung gemacht werden. Wir ergänzen das Programm wie folgt:

```

5 INPUT "X1=";A : INPUT "X2=";B
10 FOR X=A TO B STEP 0.5
35 NEXT X
40 GOTO 5
```

Die Zeilen 5 und 35 kommen neu hinzu, die Zeilen 10 und 40 werden geändert. Das ganze Programm sieht dann so aus:

```

5 INPUT "X1=";A : INPUT "X2=";B
10 FOR X=A TO B STEP 0.5
20 Y=2*X^3+3*X-2.5
30 PRINT "Y=";Y
35 NEXT X
40 GOTO 5
```

Wird nun das Programm gestartet, so fragt es nach den Werten X1 und X2 und gibt dann nach jedem ENTER alle Funktionswerte zwischen X1 und X2 in 0.5er Schritten aus. Wichtig: Bei positivem STEP muß immer die kleinere Zahl zuerst eingegeben werden.

Erklärung des Programms:

Die Zeilen 10 und 35 bilden zusammen eine FOR-NEXT Schleife, die auf die Variable X bezogen ist. Dabei ist A der Anfangswert, B der Endwert und die Angabe STEP die Schrittweite. Diese Werte können entweder mit einer Variablen, also A,B oder K usw. oder auch mit festen Zahlen belegt werden. Die Angabe STEP kann auch entfallen, die Schrittweite ist dann automatisch 1 (siehe Seite 133 im Handbuch). Die Zeile 5 dient nur zur Eingabe von A und B.

Ändern des Anzeigeformat:

Bei einigen Funktionen kann es geschehen, daß bei der Anzeige des Y-Wertes der Exponent nicht oder nur unvollständig angezeigt wird. Dann muß mittels der USING-Funktion die Zahl der angezeigten Stellen so reduziert werden, daß der Wert auf die Anzeige paßt. Hierzu kann allgemein folgende Programmzeile Verwendung finden:

```
29 USING "&&#.#####^"
```

Diese Zeile erlaubt die Ausgabe von zwei Textstellen sowie einer Fließkommazahl mit 6 Nachkommastellen und einem Exponenten. Eine genaue Erklärung dieses Befehls ist im Handbuch Seite 204 zu finden. ACHTUNG: Diese Funktion kann, falls sie nach Programm-Ende nicht wieder mit "USING" abgeschaltet wurde, Auswirkungen auf die Anzeige von Ergebnissen anderer Programme haben!

Schlußbemerkung:

Wir hoffen, daß dem Leser dieses Abschnittes nun das Grundsätzliche des Programmierens von einfachen Funktionen und Formeln etwas verständlicher geworden ist. Natürlich gibt es da noch vieles zu erklären, was aber den Rahmen dieser Einführung sprengen würde. Wir verweisen daher auch auf die Seiten 102 bis 208 des Handbuches, wo die Befehle gut verständlich erklärt werden. Ein Durchlesen dieser Seiten kann viel weiterhelfen. Wer sich mit Literatur versorgen will sei auf folgende Bücher hingewiesen:

BASIC Programmieren für Anfänger
BI-Hochschultaschenbuch 744
ISBN 3-411-00744-3 ca. 20 DM

Programmiersprachen im Griff
Band 3 BASIC
BI-Hochschultaschenbuch 797
ISBN 3-411-00797-4 ca. 25 DM

Ansonsten hilft auch das Fragen bei Kollegen im Semester weiter. Denn auch beim Programmieren gilt: Miteinander und nicht gegeneinander !

Befehlsabkürzungen

Wie Ihnen vielleicht schon bekannt ist, ist es mit den SHARP-PC's möglich, Basic-Befehle abzukürzen. Verwendet man diese Abkürzungen, so führt dies bei der Eingabe von langen Basic- Programmen zu einer enormen Zeitersparnis. Es ist wohl jedem begreiflich, daß man LP. wesentlich schneller als LPRINT und RA. wesentlich schneller als RANDOM eintippen kann. In der folgenden alphabetisch geordneten Tabelle sind die Befehle aufgeführt, bei denen sich eine Abkürzung lohnt.

<u>BEFEHL:</u>	<u>-----</u>	<u>ABKÜRZUNG:</u>
AREAD	-----	A.
BEEP	-----	B.
CALL	-----	CA.
CHR\$	-----	CH.
CLEAR	-----	CL.
CONT	-----	C.
CSAVE	-----	CS.
DATA	-----	DA.
DEGREE	-----	DE.
DIM	-----	D.
GOTO	-----	G.
INKEY\$	-----	INK.
INPUT	-----	I.
LEFT\$	-----	LEF.
LIST	-----	L.
LLIST	-----	LL.
LPRINT	-----	LP.
MEM	-----	M.
MID\$	-----	MI.
NEXT	-----	N.
PAUSE	-----	PAU.
PEEK	-----	PE.
PRINT	-----	P.
RADIAN	-----	RAD.
RANDOM	-----	RA.
RESTORE	-----	RES.
RE TURN	-----	RE.
RIGHT\$	-----	RI.
RUN	-----	R.
STOP	-----	S.
STR\$	-----	STR.
THEN	-----	T.
TRON	-----	TR.
USING	-----	U.
WAIT	-----	W.

WICHTIG: Der Punkt am Ende der Abkürzungen darf auf keinen Fall vergessen werden, da sonst der abgekürzte Befehl vom Rechner nicht erkannt wird. Obwohl die Abkürzung von STR\$ genauso lang ist wie der Befehl selbst, wurde sie mit in die Tabelle aufgenommen, da man «.» doch schneller tippen kann als «SHIFT» «\$».

Neue Befehle

Es ist mehr oder weniger bekannt, daß die im Handbuch beschriebenen Befehle längst noch nicht alle sind, die SHARP-PC's verstehen. Die folgenden Befehle werden vom Rechner zusätzlich verarbeitet:

```
PEEK x
POKE x,y
CALL x
CSAVE M x,y
CLOAD M x
```

Funktion:

PEEK x : Wie fast jedem bekannt ist, ist ein Speicher aus vielen einzelnen Speicherzellen aufgebaut, wobei in jeder Zelle eine Zahl zwischen 0 und 255 steht. Mit dem Befehl PEEK x wird es nun möglich, aus der Zelle x die Zahl zu lesen, die in ihr steht. (x zwischen 0 u. 65535). Z.B. A = PEEK 2000 - hier wird der Variablen A der Wert der Speicherzelle 2000 zugewiesen.

POKE x,y : Mit POKE x,y kann man den Wert y in die Speicherzelle x schreiben, wobei y nicht negativ und nicht » 255 sein darf. Speicherzelle x muß im RAM- Bereich sein. Der POKE-Befehl kann mit PEEK x überprüft werden. Z.B. POKE 14341,55 - PEEK 14341 - 55 erscheint.

CALL x : Mit CALL x wird eine Maschinenspracheroutine aufgerufen. Diese kann sowohl eine selbst geschriebene, als auch eine interne Routine des Rechners sein. Für x wird die Zahl eingegeben, ab der Adresse die Maschinenroutine beginnt.

CSAVE M x,y : Durch den Befehl CSAVE M x,y wird einem die Möglichkeit gegeben, Maschinensprach-Programme auf Band zu speichern oder mit einem Kommunikationsinterface in einen anderen Rechner zu übertragen. Für x wird die Startadresse und für y die Endadresse eingegeben. Z.B. CSAVE M 8192,8300 speichert den Speicherbereich von 8192 bis 8300 auf Band. Wie beim normalen CSAVE kann auch hier ein Name zugewiesen werden: CSAVE M "NAME";x,y

CLOAD M x : Mit CLOAD M x kann ein mit CSAVE M x,y abgespeichertes Maschinensprachprogramm wieder eingelesen werden. Das x stellt hierbei wieder die Startadresse dar. Die Endadresse ergibt sich automatisch. Z.B. CLOAD M 8192 Selbstverständlich ist auch hier ein Name in der Form CLOAD M "NAME";x zugelassen.

Bei allen oben genannten Befehlen ist die Zahleneingabe auch hexadezimal möglich, indem der Zahl ein & voraus geht: Z.B. PEEK &2A1F statt dezimal PEEK 10783.

Passwort

Die Rechner PC 1401/02 bieten die Möglichkeit, die eingetippten Programme durch ein PASS-Wort vor unberechtigtem Zugriff zu schützen. Ein so geschütztes Programm kann nicht mehr gelistet werden. Da es aber immer wieder vorkommt, daß das PASS-Wort vergessen wird, folgt hier eine kurze Beschreibung, wie ein unbekanntes PASS-Wort ermittelt werden kann.

Das Passwort ist im ASCII-Code seiner Buchstaben abgespeichert. Schaut man also mittels PEEK XXXXX in die entsprechenden Speicherzellen, so bekommt man damit den ASCII-Code des Zeichens der entsprechenden Speicherzelle.

1. Zeichen des Passworts steht in Speicherzelle 22209
2. Zeichen des Passworts steht in Speicherzelle 22210
3. Zeichen des Passworts steht in Speicherzelle 22211
4. Zeichen des Passworts steht in Speicherzelle 22212
5. Zeichen des Passworts steht in Speicherzelle 22213
6. Zeichen des Passworts steht in Speicherzelle 22214
7. Zeichen des Passworts steht in Speicherzelle 22215

Danach wandelt man den ASCII-Code mit CHR\$ ASCII-Code in das entsprechende Zeichen um. Beides zusammen: CHR\$ PEEK XXXXX

Beispiel:

PASS "PC-1401"

(Es muß ein Progr. im Rechner sein sonst erfolgt ein ERROR !)

Wir nehmen nun an, daß das Passwort unbekannt sei und wir ermitteln das gültige Passwort:

Befehl:	Anzeige:
CHR\$ PEEK 22209 (ENTER)	P
CHR\$ PEEK 22210 (ENTER)	C
CHR\$ PEEK 22211 (ENTER)	-
CHR\$ PEEK 22212 (ENTER)	1
CHR\$ PEEK 22213 (ENTER)	4
CHR\$ PEEK 22214 (ENTER)	0
CHR\$ PEEK 22215 (ENTER)	1

Passwort: PC-1401

Sonderfall:

Erscheint nach einer dieser Eingaben ein leeres Display (LD), so bedeutet dies, daß entweder ein SPC im Passwort verwendet wurde, oder daß in dieser Speicherzelle eine 0 steht. Gibt man nun PEEK XXXXX anstatt CHR\$ PEEK XXXXX ein, so erscheint 32 oder 0. Erscheint 32, so wurde im Passwort ein SPC verwendet - erscheint 0, endet hier das Passwort.

Beispiel:

PASS "A*B**" (* = SPC)

Befehl:	Anzeige:
CHR\$ PEEK 22209 (ENTER)	A -----> A
CHR\$ PEEK 22210 (ENTER)	LD -->PEEK 22210 (ENTER) 32 --> *
CHR\$ PEEK 22211 (ENTER)	B -----> B
CHR\$ PEEK 22212 (ENTER)	LD -->PEEK 22212 (ENTER) 32 --> *
CHR\$ PEEK 22213 (ENTER)	LD -->PEEK 22213 (ENTER) 32 --> *
CHR\$ PEEK 22214 (ENTER)	LD -->PEEK 22214 (ENTER) 0 --> Ende

Passwort: A*B** (* = SPC bzw. Leerzeichen)

In Speicherzelle 22215 steht noch der ASCII-Code des 7. Zeichens eines früher einmal eingegebenen Passworts, welches aber nicht mehr von Interesse ist (steht nach der 0).

Druckerausgabe:

1. Drucker aktivieren mit (SHIFT) (P<-->NP)
2. CHR\$ PEEK 22209 (ENTER)
3. CHR\$ PEEK 22210 (ENTER)
4. CHR\$ PEEK 22211 (ENTER)
5. CHR\$ PEEK 22212 (ENTER)
6. CHR\$ PEEK 22213 (ENTER)
7. CHR\$ PEEK 22214 (ENTER)
8. CHR\$ PEEK 22215 (ENTER)

Tips zum Tippen

Da viele SHARP-PC-Besitzer mit diesem Gerät ihre ersten größeren Erfahrungen mit Computern machen, geben wir hier folgende Hinweise zum Eintippen der Programme:

Bei der Eingabe der Programme keine Leerzeichen zwischen den Befehlen eingeben! Der Rechner ergänzt diese von selbst, wenn dies notwendig ist. AUSNAHME: In Texten, die angezeigt oder ausgedruckt werden sollen, müssen die Leerzeichen natürlich eingegeben werden. Kennzeichen für diese Texte ist, daß sie zwischen zwei Anführungszeichen stehen.

Weiterhin sollte man zur Eingabe der Programme möglichst die Kurztasten (mit SHIFT) benutzen. Für Befehle die es nicht als Kurztaste gibt, kann die entsprechende Abkürzung gegeben werden (siehe "BEFEHLSABKÜRZUNGEN"). Am Schluß jeder Zeile ENTER drücken! Zum Thema Tippfehler sei auf das SHARP-Handbuch verwiesen.

Die meisten Programme haben am Anfang einen Buchstaben in Anführungszeichen stehen. Dadurch kann das Programm über die DEF-Taste gestartet werden. Man sollte jedoch unbedingt vermeiden, diese Buchstaben doppelt zu definieren, da es sonst zur Zerstörung von Programmen im Speicher kommen kann. Deshalb erst nachsehen, ob der Buchstabe nicht schon vergeben ist! Wer da Schwierigkeiten hat, sollte die mitgelieferte Tastaturschablone verwenden (Kleines Plastikplättchen mit Löchern, kann mit weichem Bleistift beschriftet werden). Der Start über die DEF-Taste erfolgt so:

Erst den Rechner in den RUN-Mode bringen, dann DEF drücken und zuletzt den entsprechenden Buchstaben. Der Rechner beginnt dann mit dem Programm, das auf den Buchstaben folgt (ENTER ist nicht mehr nötig).

Weiterhin muß beachtet werden, daß keine Zeilennummern im Programm doppelt belegt werden dürfen. Da sonst die alte Zeile, die die Zeilennummer bereits hatte, zerstört wird. Gegebenenfalls muß ein neues Programm mit geänderten Zeilennummern eingegeben werden.

ACHTUNG: Auch die Nummern der THEN's, GOTO's und GOSUB's mit ändern, sonst läuft das Programm nicht! Auch sollten die Programme gegeneinander gesichert sein - d.h. entweder den Befehl END am Schluß, oder aber ein GOTO zum Anfang des Programmes. In letzterem Falle muß dann das Programm mit der BRK-Taste beendet werden.

Noch ein Tip:

Wer weder Zeit noch die Nerven hat, die Programme alle einzutippen, der kann auch eine entsprechende Programmkassette erwerben. Auf dieser Compact-Cassette sind alle Programme dieses Heftes abgespeichert und mit einem Cassettenrekorder und dem Cassetteninterface kann man dann die Programme bequem in den SHARP-PC einladen.

Mini-Menü

Bekanntlich hat der SHARP-PC 18 Tasten, über die Funktionen oder Programme mittels DEF aufgerufen werden können. Es gibt aber immer wieder Fälle, bei denen diese 18 Tasten nicht ausreichen. Doch dafür gibt es Abhilfe mit folgenden Programmzeilen:

```
1 BEEP(1) : INPUT "WELCHES PGM ?"; A
2 ON A GOTO 1000, 1500, 2000, 5000, 8970
3 BEEP(1) : PAUSE "FALSCH EINGABE" : GOTO 1
```

Dabei werden in Zeile 2 nach ON A GOTO hintereinander die Zeilennummern der Anfangszeile der einzelnen Programme eingegeben (z.B. 1000,1500,2000 usw.). Wird nun das Programm gestartet, so erscheint 'WELCHES PGM?' und man gibt eine Zahl gefolgt von ENTER ein. Hätte man in diesem Beispiel eine 1 eingegeben, so würde das Programm, das ab Zeile 1000 steht gestartet, bei 2 das Programm ab 1500 usw. Auf diese Weise können bis zu ca. 20 Programme aufgerufen werden, d.h. Zahlen zwischen 1 und 20.

Damit man nun weiß, welches Programm sich hinter welcher Nummer verbirgt, sollte man eine Liste auf ein Klebeetikett schreiben und dies dann in den Schutzdeckel des Rechners kleben!

Programmiertips

Es kann mitunter nützlich sein, wenn man im CAL-Modus errechnete Werte in ein Basic-Programm übernehmen kann. Hierzu folgender Hinweis:

Der Taschenrechner-Speicher M befindet sich in den Speicherplätzen 18088 bis 18094. Mittels PEEK-Befehl kann man sich nun den Wert holen und im Basic-Programm weiterverarbeiten. Natürlich kann man auch umgekehrt mit POKE Werte in diesen Speicher einschreiben. Das Feld der Möglichkeiten an dieser Stelle ist groß.

```
=====
  T I P
=====
```

Ist ein Passwort in einen Rechner eingegeben, so kann es mit POKE 18137,N (N<31) aus- und mit POKE 18137,N (N>31) wieder eingeschalten werden.

```
=====
  T I P
=====
```

Dieses Programm zählt ziemlich genau die vergangene Zeit in Minuten. Wird es z.B. um 12.00 Uhr gestartet, so wird um 12.30 Uhr vom Rechner 30 und um 13.10 vom Rechner 70 angezeigt.

Listing:

```
10 WAIT 3524 : A=0
20 PRINT A : A=A+1 : GOTO 20
```

=====
T I P
=====

Der PC-1401 schaltet sich ja bekanntlich nach einer gewissen Zeit von selbst ab. Wer dies beschleunigen möchte, muß nur CALL 1456 eintippen.

=====
T I P
=====

Es steht zwar im Handbuch, aber einige wissen es doch noch nicht: Drückt man zusammen mit ALL RESET die rote C-CE Taste, so bleiben die Programme im Rechner erhalten !!!

=====
APRIL - TIP
=====

Dies kurze Maschinenprogramm erzeugt einen sehr hohen Ton. Nach der Eingabe mit POKE ... muß es mit CALL &45D0 gestartet werden. Es ist dabei zu beachten, daß der Rechner nur mit ALL RESET abgeschaltet werden kann (nicht mit dem EIN/AUS-Schalter). Aus diesem Grund eignet sich das Programm besonders um Freunde und Bekannte zu ärgern, die ebenfalls einen 1401 besitzen.

POKE 17872,18,95,97,16,223,96,239,223,121,69,208
CALL &45D0

=====
T I P
=====

Ihr SHARP-PC ist mit den meisten seiner Brüder softwarekompatibel (zumindest hinsichtlich der normalen Rechen- und Text-Befehle), so daß z.B. die Programme des PC 1245 sämtlich auch auf dem PC 1401/02/03/04 laufen. Bei anderen wie z.B. PC 1211 muß berücksichtigt werden, daß dieser eine 24-stellige Anzeige hat und damit die Texte und Hinweise in ihrer Länge evtl. gekürzt werden müssen.

=====
T I P
=====

Für alle die es noch nicht gemerkt haben: Mit dem SHARP-PC kann man auch während der Eingabe rechnen. Das bedeutet: Fragt z.B. ein Programm nach der Variablen "X=" und man müßte den Wert für $2\pi-2.6$ eingeben, so tippen wird während das Programm läuft einfach $2*\pi-2.6$ «ENTER». Das Programm setzt jetzt für die Variable X das Ergebnis dieser Rechnung ein.

=====
T I P
=====

Es wird oft behauptet, daß mit den SHARP-PC's ein Umrechnen von Grad in Bogenmaß (Radiant) im Taschenrechnermodus (CAL- Mode) nicht möglich sei. Dies ist nicht richtig! Im Handbuch steht zwar nichts darüber, aber mit folgendem Trick ist es leicht möglich.

Angenommen, wir wollen 34 Grad in Bogenmaß umrechnen, so geben wir im CAL-Modus folgendes ein:

34 SIN	Sinus von 34 Grad
SHIFT DRG	Umschalten auf Radiant
SHIFT SIN-1	Arcussinus zurück auf den Winkel

und schon haben wir das Ergebnis in Radiant. Umgekehrt kann man so natürlich auch von Radiant in Grad oder von Grad in Neugrad usw. umrechnen. Es muß dann nur der richtige Modus mit der DRG - Taste gewählt werden.

=====
T I P
=====

In einer Standardtextvariablen können ohne Dimensionierung nur 7 Zeichen gespeichert werden. Nimmt man aber statt A\$ z.B. AA\$, so können mehr Zeichen abgespeichert werden.

Die Standardvariablen stehen ab 17872 (&45D0) bis 18079 (&469F) im Speicher. Jede Variable hat eine Länge von 8 Byte, wobei sie in der Reihenfolge Z,Y,X...B,A abgespeichert sind (z.B. Z von 17872-17879). Der Speicher des CAL-Mode steht von 18088-18095 im Rechner.

Beim PC-1401/02 ist keine Möglichkeit vorgesehen, Werte direkt vom CAL-Mode in Basicprogrammen unterzubringen (bis auf Statistik-Ergebnisse). Nachfolgendes Programm weist der Variablen M, den Wert des Speichers "M" des CAL-Mode zu.

Listing:

```
10 FOR N=17976 TO 17983:POKE N,PEEK(N+112):NEXTN
20 PRINT M
```


Kapitel 4.1

Mathematik

Kurvendiskussion

Beschreibung:

Programm zur Berechnung von Nullstellen einer gegebenen Funktion, sowie zur Durchführung einer Polynomdivision und der Flächenberechnung unter der Kurve (Integral).

Die Programme sind alle ohne absolute Sprungbefehle geschrieben, d.h. sie sind beliebig im Speicher verschiebbar ohne die Zeilennummern anpassen zu müssen. Dies erhöht die Kompatibilität und Übersichtlichkeit erheblich. Leider werden die Programme dadurch etwas langsamer und länger. Wer meint, er wird nie etwas an seinen Programmen ändern, der kann die Sprunglabels (Goto "XXX" bzw Gosub "XXX" in Goto bzw. Gosub 'Zeilennummer') abändern. Ich rate jedoch davon ab. Nicht änderbar sind alle Sprünge nach "F0" und "F1".

Noch ein Tip am Rande: $\text{Print int}(x * 1e3 + .5) / 1e3$ gibt x auf 3 Stellen gerundet aus. In den Programmen wird verschiedentlich davon Gebrauch gemacht, um einen Wert vollständig auf die Anzeige zu bringen. (Ich besitze einen PC 1401 mit 16 stelliger Anzeige). Dies gilt nur für Ausgaben über den Print-Befehl, nicht jedoch für Wertzuweisungen wie $y = \text{int}(\dots)/\dots$; hier darf nichts geändert werden.

Funktionen

FU enthält die zu berechnende Funktion für die Programme Nullstellen und Integral. In der Zeile vor der Funktion oder in der gleichen Zeile muß "F0", nach der Funktion oder in der nächsten Zeile muß ein 'Return' stehen. Im Unterprogramm F1 wird die erste Ableitung zur Funktion unter F0 berechnet. Es gilt das gleiche wie für F0.

Bsp.: "F0" $Y = \sin(x) * \cos(x) : \text{return}$

Nullstellen

Das Programm berechnet die Nullstellen einer Funktion nach dem Algorithmus des Regula Falsi.

Anzeige	Eingabe	Beschreibung
F:	F0 F1	Berechnung der Nullstellen von $f(x)$ Berechnung der Nullstellen von $f'(x)$
XU = XO =	Zahl Zahl	Eingabe der unteren und oberen Grenze des 'Suchintervalles'
SW =	Zahl	Schrittweite. Intervall in dem die Funktion die x-Achse nur einmal schneiden darf. (norm: sw = 1)

Nach Ablauf des Programms sind die gefundenen Nullstellen in der Feldvariablen $n(i)$ gespeichert. D.h., die erste Nullstelle steht in dem Feld $n(0)$, die zweite in dem Feld $n(1)$ usw. Diese können dann für eigene Rechnungen weiterverwendet werden. z. B. $4 * n(1) + 2 * n(0)$ 'enter'.

Bsp.: Berechnung der Nullstelle von x^2-3 .

```

y = x^2-3           unter FU eintragen

Eingabe:  F0           für Funktion berechnen
Anzeige:  xu =
Eingabe:  -5           untere Intervallgrenze -5
Anzeige:  xo =
Eingabe:  5           obere Intervallgrenze +5
Anzeige:  sw =
Eingabe:  1           Schrittweite 1
Anzeige:  -1.732050808 'enter'
Anzeige:  1.732050808  'enter'
Eingabe:  >n(0)       'enter'
Anzeige:  -1.732050808

```

Polynomdivision

Bei der Abfrage "Div / Mult / Quad " bitte nur

```

d --- Division
m --- Multiplikation
q --- quadrieren

```

eingeben. Im folgenden wird nur die Polynomdivision beschrieben. Die Multiplikation läuft dann analog ab. Ebenso das Quadrieren, welches eigentlich nur die Multiplikation einer Funktion mit sich selbst darstellt, weshalb die zweite Eingabe der Funktion überflüssig wird.

gegeben: $f(x) = (8x^5 - 3x^3 - 24x^2 - 28) / (2x^3 + x^2 - 7)$

Nach dem Programmstart erscheint in der Anzeige "G1: ". Hier wird der Grad des ersten Polynoms eingegeben, also 5. Anschließend werden die Koeffizienten des ersten Polynoms (bezeichnet mit A) eingegeben. In der Anzeige erscheint immer erst die Nummer des Koeffizienten, welche man erst mit 'enter' quittieren muß um fortfahren zu können:

```

Anzeige:  A * X ^ 5   'enter'
Anzeige:  ?           Eingabe von A (hier 8)
Anzeige:  A * X ^ 4   'enter'
Anzeige:  ?           Eingabe von A (hier 0)
...

```

Die Eingabe des zweiten Polynoms B mit dem Grad G2 erfolgt nach dem selben Prinzip.

Nach der Eingabe von Bx^0 erscheint nach kurzer Zeit das Ergebnis, welches folgendermaßen interpretiert wird:

$4x^2 - 2x - 0.5$ Rest: $4.5x^2 - 14x - 31.5$

d.h.:
$$4x^2 - 2x - 0.5 + \frac{4.5x^2 - 14x - 31.5}{2x^3 + x^2 - 7}$$

Quittiert man die Eingabe eines Koeffizienten nur mit 'enter' so wird die Eingabe abgeschlossen und das Programm fährt mit der Eingabe des zweiten Polynoms oder der Berechnung fort.

$$(x^7-6x^5)/(x-1)$$

...
 Anzeige: A * X ^ 5 'enter'
 Anzeige: ? 'enter'
 Anzeige: G2: ...

Mult: $(3x^5+4x^2-7x) * (3x^2+5) = 9x^7+15x^5+12x^4-21x^3+20x^2-35x$

Quad: $(3x^2+5x-7)^2 = 9x^4+30x^3-17x^2-70x+49$

Integral

Dieses Programm berechnet die Fläche unter der Kurve F0 im Intervall x_u, x_o . Um richtige Ergebnisse zu erhalten muß die Kurve von Nullstelle zu Nullstelle integriert werden und die errechneten Flächen absolut addiert werden.

Bsp.: $y = \int_{-1}^1 x \, dx = 0$ falsch

$$y = \left| \int_{-1}^0 x \, dx \right| + \left| \int_0^1 x \, dx \right| = |-0.5| + |0.5| = 1$$

Das Programm rechnet mit der Simpson Formel. Deshalb kann man bei n die Anzahl der Stützstellen frei wählen. Bei $n=20$ erhält man für jede Funktion die Fläche auf 10 Stellen genau. Diese Abfrage kann deshalb entfallen und durch $n=20$ ersetzt werden. (im Programm ändern)

Variablenbelegung der Programme

Beim Aufruf einiger Programme werden alle Variablen gelöscht!
 Hier werden nur die nach dem Programmablauf bereitstehenden, verwertbaren Variablen aufgezählt.

Funktionen:	y d	Funktionswert von f(x) bzw f'(x) Delta X für die Näherung.
Nullstellen:	n(0)..n(i)	gefundene Nullstellen
Integral:	i	Flächeninhalt
PQ - Formel:	w a,b	Ergebnis der Wurzel Ergebnisse der quadratischen Gl.
Polynomdivision:	c(n) e	Ergebnispolynom (ganzzrational) Grad des Ergebnispolynoms
div:	a(n) b(n)	Restpolynom (Zähler) Restpolynom (Nenner)
mult,quad:	a(n) b(n)	Polynom A Polynom B

Listing:

```

1 "F0" Y=sin x:return
2 "F1" d=ten-7:x=x+d:gosub "F0":z=y:x=x-d:gosub "F0":
  y=(z-y)/d:return
5 rem Nullstellen
10 "a"clear:dim n(5):input "f: ";f$, "xu = ";a, "xo = ";
  b, "sw = ";d
15 "NS" gosub "SO":if su beep 1:print x:n(i) = x:i=i+1:
  a=x+d:goto "NS"
20 end
25 "SO" su=0:x=a:gosub f$:ly=y:lx=x:x=x+d
30 if ly=0 let su=1:x=x-d:return
35 "L1" gosub f$:if y*ly<=0 let su=1:if y<>0 gosub "RF"
40 if y*ly>0 and not su let ly=y:lx=x:x=x+d
45 if x<=b and not su then "L1"
50 return
55 "RF" z=x:x=x-y*(x-lx)/(y-ly):lx=z:ly=y:gosub f$:
  if abs(lx-x) > ten-6 then "RF"
60 return
65 rem Polynomdivision
70 "z"clear:input "Div/Mult/Quad ";w$, "zg: ";z:
  dim a(z)
75 for i=z to 0 step -1:print "a * x ^ ";i:input a(i):
  next i
77 if w$="q" let w$="m":n=z:dim b(n):for i=0 to z:
  b(i)=a(i):next i:goto "19"
80 input "ng: ";n:dim b(n):for i=n to 0 step -1:
  print "b * x ^ ";i:input b(i):next i
85 "19" w$=w$+"+":gosub w$:for i=e to 0 step -1:
  if c(i)<>0 print c(i);"*x^";i
90 next i:if w$="m+" or a(z)=0 end
95 print "Rest:":for i=n-1 to 0 step-1:
  if a(i)<> 0 print a(i);"*x^";i
100 next i:end
105 "D+" e=z-n:dim c(e):for i=e to 0 step -1:
  c(i)=a(z)/b(n):z=z-1
110 if c(i)<>0 for j=z to z-n+1 step -1:
  a(j)=a(j)-c(i)*b(j-i):next j
115 next i:return
120 "M+" e=z+n:dim c(e):for i=0 to z:for j=0 to n:
  c(i+j)=c(i+j)+a(i)*b(j):next j:next i:return
145 rem integral
150 "s"a=0:input "xu = ";x, "xo = ";e, "n = ";n:
  h=(e-x)/2/n:gosub "f0":a=a+y
155 for i=1 to n:x=x+h:gosub"f0":a=a+4*y:x=x+h:
  gosub"f0":a=a+y+y:next i:a=(a-y)*h/3:print a:end
160 rem pq-Formel
170 "x"input "ax^2 / a=";a, "px / p = ";p, "q = ",q
175 p=p/a:q=q/a:w=p^2/4-q:if w<0 print "wurzel < 0":end
180 a=-p/2-sqrw:b=-p/2+sqrw:print a:print b:end
9999 rem M -> Var
10000 ", " AREAD J$:FOR I=0 TO 6:
  POKE(90-ASC J$)*8+17872+I,PEEK(18088+I):NEXT I:END
10005 rem Anz -> M
10010 " " AREAD J:FOR I=0 TO 6:POKE 18088+I,PEEK(18000+I):
  NEXT I:END

```

Polynomdivision I

Beschreibung:

Hier wird aus den Polynomen $P(x)$ und $Q(x)$ das Quotienten-Polynom $X(x)$ sowie das Restpolynom $R(x)$ berechnet, so daß gilt: $P(x)=Q(x)*X(x)+R(x)$.

Listing:

```

100 REM POLYNOMDIVISION
110 PAUSE "DIVIDENTEN-POLY:"
120 INPUT "GRAD ? ";N
130 DIM P(N),Y(N)
140 FOR I=0 TO N
150 PAUSE "X(";I;") : "
160 INPUT "? ";P(I)
170 NEXT I
180 PAUSE "DIVISOR-POLY:"
190 INPUT "GRAD ? ";M
200 IF M>N THEN 190
210 DIM Q(M)
220 FOR I=0 TO M
230 PAUSE "X(";I;") : "
240 INPUT "? ";Q(I)
250 NEXT I
260 R=N-M
270 FOR I=N TO M STEP-1
280 Y(I-M)=P(I)/Q(M)
290 FOR J=0 TO M
300 K=I-J
310 P(K)=P(K)-Y(I-M)*Q(M-J)
320 IF ABS P(K)>= ABS (Y(I-M)*Q(M-J))*1E-7 THEN 340
330 P(K)=0
340 NEXT J:NEXT I
350 PAUSE "QUOTIENT-POLY:"
360 FOR I=0 TO R
370 PRINT "X(";I;")= ";Y(I)
380 NEXT I
390 PAUSE "RESTPOLYNOM:"
400 IF P(R)<>0 THEN 420
410 R=R-1:GOTO 400
420 FOR I=0 TO R
430 PRINT "R(";I;")= ";P(I)
440 NEXT I
450 END

```

Polynomdivision II

Beschreibung:

Beispiel (1) : $(x^3 - 2x^2 - 5x + 6) : (x - 1) = x^2 - x - 6$

$$\begin{array}{r} x^3 - 2x^2 - 5x + 6 \\ \underline{-x^2 + x} \\ -6x + 6 \\ \underline{-6x + 6} \\ 0 \end{array}$$

Das reduzierte Polynom lautet $x^2 - x - 6$

Beispiel (2) : $(x^5 - 3x^4 + 2x^3 - 7x^2 + 6x + 5) : (x^3 - 2x^2 + 5x + 2) = x^2 - x - 5$

$$\begin{array}{r} x^5 - 3x^4 + 2x^3 - 7x^2 + 6x + 5 \\ \underline{-x^4 + 2x^3 - 5x^2 - 2x} \\ -5x^3 - 4x^2 + 8x + 5 \\ \underline{-5x^3 + 10x^2 - 25x - 10} \\ -14x^2 + 33x + 15 \end{array}$$

Das reduzierte Polynom lautet

$$x^2 - x - 5 + \frac{-14x^2 + 33x + 15}{x^3 - 2x^2 + 5x + 2}$$

Programmablauf am Beispiel (1) :

1. Grad des Zählers : 3
2. Grad des Nenners : 1

3. Koeffizienten des Zählers : $a_3 = 1$
 $a_2 = -2$
 $a_1 = -5$
 $a_0 = 6$

4. Koeffizienten des Nenners : $b_1 = 1$
 $b_0 = -1$

5. Nach der Berechnung erfolgt die Ausgabe : $a_2 = 1$
 $a_1 = -1$
 $a_0 = -6$

Bei der Ausgabe werden die Koeffizienten des Ergebnispolynoms angezeigt. Um mit der Ausgabe fortzufahren muß ENTER gedrückt werden. Falls es sich um eine unecht gebrochenrationale Funktion handelt, wird die Funktion in eine Summe aus Polynom und einer echt gebrochenrationalen Funktion zerlegt. (siehe Bsp. (2)) Bei der Ausgabe erscheint zusätzlich der Punkt "REST:". Hier werden dann die Koeffizienten des Zählers der echt gebrochenrationalen Funktion angezeigt.

Listing:

```

10 DIM A(20),B(20),C(20)
20 "A"
30 PRINT "*** POLYNOM DIVISION ***"
40 INPUT "ZAEHLERGRAD:";GA
50 INPUT "NENNERGRAD:";GB
60 IF GA<GB THEN GOTO "A"
70 WAIT 45
80 PAUSE "ZAEHLERKOEFFIZIENTEN"
90 FOR I=GA TO 0 STEP-1
100 PRINT "a";I;"="
110 INPUT A(I)
120 NEXT I
130 PAUSE "NENNERKOEFFIZIENTEN"
140 FOR I=GB TO 0 STEP-1
150 PRINT "b";I;"="
160 INPUT B(I)
170 NEXT I
175 PAUSE "BERECHNUG"
180 FOR I=GA TO GB STEP-1
190 C(I-GB)=A(I)/B(GB)
200 FOR J=I TO I-GB STEP-1
210 A(J)=A(J)-C(I-GB)*B(GB-(I-J))
220 NEXT J
230 NEXT I
240 WAIT
250 PRINT "ERGEBNISPOLYNOM"
260 FOR I=GA-GB TO 0 STEP-1
270 PRINT "a";I;"=";C(I)
280 NEXT I
290 F=0
300 FOR I=0 TO GA
310 IF A(I)<>0 THEN LET F=1
320 NEXT I
330 IF F=1 GOSUB "REST"
335 WAIT 120
340 PRINT "FERTIG":WAIT:GOTO "A"
350 "REST"
360 PRINT "REST : "
370 FOR I=GA TO 0 STEP-1
380 PRINT "a";I;"=";A(I)
390 NEXT I
400 RETURN

```


Polynommultiplikation

Beschreibung:

Analog zur Polynomdivision werden hier zwei Polynome $P(x)$ und $Q(x)$ multipliziert. Wenn die Polynome $P(x)$ vom Grad N und $Q(x)$ vom Grad M sind, so ist das Produkt-Polynom vom Grad $N+M$.

Listing:

```
100 REM POLYNOMMULTIPLIKATION
110 INPUT "GRAD P(X) ? ";N
120 DIM P(N)
130 FOR I=0 TO N
140 PAUSE "X(";I;") : "
150 INPUT "? ";P(I)
160 NEXT I
170 INPUT "GRAD Q(X) ? ";M
180 DIM Q(M),Y(N+M)
190 FOR I=0 TO M
200 PAUSE "X(";I;") : "
210 INPUT "? ";Q(I)
220 NEXT I
230 FOR I=0 TO N+M
240 Y(I)=0:NEXT I
250 FOR I=0 TO N
260 FOR J=0 TO M
270 Y(I+J)=Y(I+J)+P(I)*Q(J)
280 NEXT J:NEXT I
290 PAUSE "PRODUKT-POLYNOM:"
300 FOR I=0 TO N+M
310 PRINT "X(";I;") = ";Y(I)
320 NEXT I
330 END
```

Interpolation nach Lagrange

Beschreibung:

(Näherungsdarstellung von Funktionen mit der Interpolationsformel nach LAGRANGE)

Das Programm LAGRANGE bestimmt durch Eingabe von 6 Wertepaaren (X1,Y1).. (X6,Y6) eine Näherungsfunktion $y=f(x)$.

Beispiel: Ermittlung des Zählerstandes eines beliebigen Videorecorders in Abhängigkeit von der Spielzeit

1. Videokassette E-180 starten (PLAY drücken)
2. nach 35 Minuten Zählerstand des Recorders notieren
3. nach 70 Minuten Zählerstand des Recorders notieren
4. 105 ..
5. 140 ..
6. 180 ..

Das Programm mit DEF A starten, und anschl. die 6 Wertepaare eingeben:

	X	Y	X=Spielzeit in Minuten Y=Zählerstand
Wertepaar 1:	(X1,Y1)	0	0
Wertepaar 2:	(X2,Y2)	35	112
Wertepaar 3:	(X3,Y3)	70	490
Wertepaar 4:	(X4,Y4)	105	1730
Wertepaar 5:	(X5,Y5)	140	3210
Wertepaar 6:	(X6,Y6)	180	4400

Näherungsformel:

$$1.881859533E-7*x^5-9.96843258E-5*x^4+1.748732612E-2*x^3$$

Anschl. kann sofort durch Eingabe eines beliebigen Wertes für die Spielzeit (X), der entsprechende Zählerstand (Y) ermittelt werden. z.B. bis zu welchem Zählerstand muß vorgespult werden, wenn ein 120 min Film aufgenommen wurde, um die restliche Bandkapazität für eine Aufnahme zu nutzen.

$$x=120(\text{min}) \quad y=f(x)=2385(\text{Zählerstand})$$

Bedienung:

- DEF A: Neustart. Eingabe der Wertepaare
 DEF S: Erneute Anzeige der Näherungsfunktion
 DEF D: Ermittlung eines Funktionswertes $f(x)=Y$ durch Eingabe von X

Das Programm wurde auf einem SHARP PC-1211 entwickelt, danach wurde es für einen SHARP PC-1403 umgeschrieben. Damit dürfte dieses Programm auf allen SHARP-Pocketcomputern lauffähig sein, da nur die notwendigsten Anpassungen an den PC-1403 vorgenommen wurden.

Listing:

```

10 "A":CLEAR:USING "##"
20 FOR U=1 TO 6:PAUSE "WERTEPAAR";U:V=32+U:
  INPUT "X=";A(V):W=38+U:INPUT "Y=";A(W)
30 NEXT U
40 USING:PAUSE "BITTE ETWAS GEDULD!"
50 S=1:T=0
60 A(45)=A(45)+1
70 A=A(34):IF A(45)>=2 LET A=A(33)
80 B=A(35):IF A(45)>=3 LET B=A(34)
90 C=A(36):IF A(45)>=4 LET C=A(35)
100 D=A(37):IF A(45)>=5 LET D=A(36)
110 E=A(38):IF A(45)=6 LET E=A(37)
120 X=A(32+A(45))
130 Y=A(38+A(45))
140 F=S
150 G=(-B-A-C-D-E)*S
160 H=(A*B+B*C+A*C+B*D+A*D+C*D+B*E+A*E+C*E+D*E)*S
170 I=(-A*B*C-A*B*D-B*C*D-A*C*D-A*B*E-B*C*E-A*C*E-
  B*D*E-A*D*E-C*D*E)*S
180 J=(A*B*C*D+A*B*C*E+A*B*D*E+B*C*D*E+A*C*D*E)*S
190 K=(-A*B*C*D*E)*S:IF T=1 GOTO 210
200 GOTO 310
210 GOSUB 410
220 IF Z=6 GOTO 240
230 GOTO 50
240 "S":BEEP 3:PRINT " ";A(27);" X^5"
250 PRINT " ";A(28);" X^4"
260 PRINT " ";A(29);" X^3"
270 PRINT " ";A(30);" X^2"
280 PRINT " ";A(31);" X^1"
290 PRINT " ";A(32);" X^0"
300 GOTO 490
310 L=X^5*F
320 M=X^4*G
330 N=X^3*H
340 O=SQU X*I
350 P=X*J
360 Q=K
370 R=L+M+N+O+P+Q
380 S=Y/R
390 T=1
400 GOTO 140
410 A(27)=A(27)+F
420 A(28)=A(28)+G
430 A(29)=A(29)+H
440 A(30)=A(30)+I
450 A(31)=A(31)+J
460 A(32)=A(32)+K
470 Z=Z+1
480 RETURN
490 "D":INPUT "X=";X
500 Y=X^5*A(27)+X^4*A(28)+X^3*A(29)+SQU X*A(30)+
  X*A(31)+A(32)
510 USING "#####.####"
520 PRINT "Y=";Y
530 GOTO 490

```

Gleichungen 3. Grades

Beschreibung:

Programm zum Berechnen von Gleichungen 3. Grades.

Gestartet wird mit DEF 'G' und es werden die Variablen A1,A2,A3 und A4 abgefragt.

Normalform der Gleichung: $A1 X^3 + A2 X^2 + A3 X + A4 = 0$

Ausgegeben werden X1,X2 und X3.

Hinweis: Es werden keine Doppelpole erkannt !

Listing:

```

340 "G"
350 PRINT "GLEICHUNGEN III. GRADES <<<<<"
360 INPUT "A1=";A1
370 INPUT "A2=";A2
380 INPUT "A3=";A3
390 INPUT "A4=";A4
400 IF A1=0 THEN PRINT "FALSCH":END
410 R=A2/A1:S=A3/A1:T=A4/A1
420 P=S-R*R/3:Q=2*R*R*R/27-S*R/3+T
430 A=-Q/2:B=A*A+P*P*P/27
440 BEEP 1
450 IF B=0 THEN B=0:F3=1
460 IF B<0 THEN GOTO 530
470 F3=1
480 B=SQR(B):FN=(A+B):GOSUB 690:U=FE
490 FN=(A-B):GOSUB 690:V=FE
500 Y1=U+V:X1=Y1-R/3
505 USING "*****.***"
510 IF F1=0 THEN PRINT "X1=";X1;" REELL !"
520 GOTO 620
530 RR=SQR(-(P*P*P/27)):FN=RR:GOSUB 690
540 X=2*FE
550 X1=X*COS((PI/3)-R/3):X2=X*COS((PI/3+2/3*PZ)-R/3)
560 X3=X*COS((PI/3+4/3*PZ)-R/3)
570 IF F1=1 THEN RETURN
580 PRINT " ALLE LOESUNGEN SIND REELL !"
585 USING "*****.***"
590 PRINT "X1=";X1
600 PRINT "X2=";X2
610 PRINT "X3=";X3
615 END
620 RT=-Y1/2:IT=(U-V)/2*SQR(3)
630 X2=RT-R/3:X3=X2
640 IF F1=1 THEN F2=1
650 IF B=0 THEN PRINT " LOESUNGEN SIND REELL !":GOTO 585
660 PRINT " X2 , X3 SIND ZU KOMPLEX"
670 PRINT "X2=";RT-R/3;" + I";IT
680 PRINT "X3=";RT-R/3;" - I";IT
685 END
690 FE=SGN(FN)*ABS(FN)^(1/3)
700 RETURN

```

Lineares Gleichungssystem

Beschreibung:

Das Programm berechnet aus 3 linearen Gleichungen die 3 Variablen X,Y,Z. Vom Rechner werden zu den Berechnungen die jeweiligen Koeffizienten der Variablen gefordert, wobei gilt:

1. Gleichung: $A_{11} * X + A_{12} * Y + A_{13} * Z = C_1$
2. Gleichung: $A_{21} * X + A_{22} * Y + A_{23} * Z = C_2$
3. Gleichung: $A_{31} * X + A_{32} * Y + A_{33} * Z = C_3$

Achten Sie bitte auf Tippfehler bei der Eingabe, sonst kommt es zu falschen Ergebnissen.

Listing:

```

410 "L": CLEAR :N=3: DIM X(9),C(3)
420 FOR I=1 TO N: PAUSE "A1";I: INPUT X(I): NEXT I
430 FOR I=1 TO N: PAUSE "A2";I: INPUT X(N+I): NEXT I
440 FOR I=1 TO N: PAUSE "A3";I: INPUT X(2*N+I): NEXT I
450 FOR I=1 TO N: PAUSE "C";I: INPUT C(I): NEXT I
470 U=X(1)*X(5)*X(9)+X(2)*X(6)*X(7)+X(3)*X(4)*X(8)
475 P=X(7)*X(5)*X(3)+X(8)*X(6)*X(1)+X(9)*X(4)*X(2)
480 D=U-P
490 U=C(1)*X(5)*X(9)+X(2)*X(6)*C(3)+X(3)*C(2)*X(8)
495 P=C(3)*X(5)*X(3)+X(8)*X(6)*C(1)+X(9)*C(2)*X(2)
500 E=U-P:K=E/D
505 U=X(1)*C(2)*X(9)+C(1)*X(6)*X(7)+X(3)*X(4)*C(3)
510 P=X(7)*C(2)*X(3)+C(3)*X(6)*X(1)+X(9)*X(4)*C(1)
515 F=U-P:G=F/D
520 U=X(1)*X(5)*C(3)+X(2)*C(2)*X(7)+C(1)*X(4)*X(8)
525 P=X(7)*X(5)*C(1)+X(8)*C(2)*X(1)+C(3)*X(4)*X(2)
530 R=U-P:T=R/D
540 PRINT "X1=";K: PRINT "X2=";G: PRINT "X3=";T: GOTO 540

```

Relativitätstheorie

Beschreibung:

Programm zu Berechnungen der Relativitätstheorie von Albert Einstein (1879-1955).

Es sind vorgesehen:

Berechnungen zur Zeit --- Start mit DEF 'A'
 Berechnungen zur Länge --- Start mit DEF 'B'
 Berechnungen zur Masse --- Start mit DEF 'C'

Die eingesetzte Konstante der Lichtgeschwindigkeit ist 299792500 m/s.

Beispiel:

Berechnungen zur Zeit:

Periode der Uhr im ruhenden System z.B. 1
 Geschwindigkeit des bewegten Systems z.B. 150000000 m/s
 Ergebnis:
 Periode der Uhr im bewegten System = 1.15496

Angewandte Formeln:

$$T' = T / \sqrt{(1 - v^2 / c^2)}$$

$$L' = L * \sqrt{(1 - v^2 / c^2)}$$

$$M' = M * \sqrt{(1 - v^2 / c^2)}$$

Listing:

```

10 C=299792500
20 PRINT "RELATIVITAETSTHEORIE"
30 PRINT "<A>BERECHNUNGEN ZUR ZEIT"
40 PRINT "<B>BERECHNUNGEN ZUR LAENGE"
50 PRINT "<C>BERECHNUNGEN ZUR MASSE"
60 END
70 "A"
80 PRINT "BERECHNUNGEN ZUR ZEIT"
90 INPUT "PERIODE(RUHE) ", P
100 INPUT "GESCHW. (BEWEGT)SYSTEM", V
110 P1=P/SQR(1-V*V/C/C)
120 PRINT "PERIODE(BEWEGT)"; P1
130 END
140 "B"
150 PRINT "BERECHNUNGEN ZUR LAENGE"
160 INPUT "LAENGE(BEWEGT)SYSTEM ", L
170 INPUT "GESCHW. (BEWEGT)SYSTEMS ", V
180 REM ----- LAENGENKONTRAKTION -----

```

```

190 L1=L*SQR(1-V*V/C/C)
200 PRINT "LAENGE(RUHE)SYSTEM ";L1
210 END
220 REM ----- BERECHNUNGEN ZUR MASSE -----
230 "C"
240 PRINT "BERECHNUNGEN ZUR MASSE"
250 INPUT "RUHEMASSE DES TEILCHENS ",M
260 INPUT "GESCHW. TEILCHEN ",V
270 REM ----- MASSEFORMEL -----
280 M1=M/SQR(1-V*V/C/C)
290 PRINT "MASSE TEILCHEN ";M1
300 END

```

Statistische Berechnungen

Beschreibung:

Dieses Programm erlaubt es 9 verschiedene Rechnungen der Statistik durchzuführen. Es ist für jeden der öfters mit statistischen Berechnungen zu tun hat eine Erleichterung.

DEF F	Fakultät einer natürlichen Zahl berechnen
DEF Z	Kombination ohne Wiederholung
DEF A	Kombination mit Wiederholung
DEF X	Variation ohne Wiederholung
DEF S	Variation mit Wiederholung
DEF B	Binominalverteilung
DEF V	Poissonverteilung
DEF H	Hypergeometrische Verteilung
DEF M	Multinomialverteilung

Listing:

```

10 "F":CLS:WAIT 1:CLEAR
15 PRINT "FAKULTAT berechnen"
20 INPUT "ZAHL ";Z
25 E=1:B=B+1:IF B>3 THEN LET B=3
30 FOR A=1 TO Z
35 E=E*A
40 NEXT A
45 CURSOR 7,B:PRINT "! = ";E
50 GOTO 20
100 "Z":CLS:CLEAR:WAIT 1:E=1:F=1
105 PRINT "Kombination OHNE Wh"
110 INPUT "n = ";N
115 INPUT "k = ";K
118 IF K=0 THEN LET E=1:GOTO 135
120 FOR A=(N-K+1)TO N
125 LET E=E*A
130 NEXT A
135 FOR A=1 TO K
140 F=F*A
145 NEXT A
150 E=E/F:WAIT
155 CURSOR 0,3:PRINT "C(oW) = ";E
160 GOTO 100
200 "A":CLS:CLEAR:WAIT 0:E=1:F=1

```

```

205 PRINT "Kombination MIT Wh"
210 INPUT "n = ";N
215 INPUT "k = ";K
218 IF K=0 THEN LET E=1:GOTO 235
220 FOR A=N TO(N+K-1)
225 LET E=E*A
230 NEXT A
235 FOR A=1 TO K
240 LET F=F*A
245 NEXT A
250 E=E/F:WAIT
255 CURSOR 0,3:PRINT "C(mW) = ";E
260 GOTO 200
300 "X":CLS:CLEAR:WAIT 0:E=1
305 PRINT "Variation OHNE Wh"
310 INPUT "n = ";N
315 INPUT "k = ";K
318 IF K=0 THEN LET E=1:GOTO 331
320 FOR A=(N-K+1)TO N
325 E=E*A
330 NEXT A
331 WAIT
335 CURSOR 0,3:PRINT "V(oW) = ";E
340 GOTO 300
350 "S":CLS:CLEAR:WAIT 0
355 PRINT "Variation MIT Wh"
360 INPUT "n = ";N
365 INPUT "k = ";K
370 LET E=N^K:WAIT
375 CURSOR 0,3:PRINT "V(mW) = ";E
380 GOTO 350
390 "B":CLS:WAIT 1:A=0:E=1:F=1
395 PRINT "BINOMINALVERTEILUNG"
400 INPUT "n = ";N
405 INPUT "p = ";P
410 INPUT "x = ";X
415 IF X=0 THEN 435
420 FOR A=(N-X+1)TO N
425 E=E*A
430 NEXT A
435 FOR A=1 TO X
440 F=F*A
445 NEXT A
450 E=E/F
455 LET E=E*(P^X)*(1-P)^(N-X):LET E=INT(E*10^5)/10^5
460 ERW=INT(N*P*10^5)/10^5
465 VAR=INT(N*P*(1-P)*10^5)/10^5
470 CLS:PRINT "n= ";N;" p= ";INT(P*10^5)/10^5;" x= ";X
475 CURSOR 0,1:PRINT "P(X= ";X;" )=";E
480 CURSOR 0,2:PRINT "E(X) = ";ERW
485 WAIT:CURSOR 0,3:PRINT "var(x) = ";VAR
490 GOTO 390
500 "V":CLS:WAIT 1:E=1:F=0
505 PRINT "POSSIONSVERTEILUNG"
510 INPUT "n*p = lamda = ";LAM
515 INPUT "x = ";X
520 FOR A=1 TO X
525 E=E*A
530 NEXT A
535 F=LAM^X*EXP-LAM

```



```

540 E=INT(F*10^5/E)/10^5
545 CLS:PRINT "lamda = ";INT(10^5*LAM)/10^5;"x = ";X
550 CURSOR 0,1:PRINT "P(X=";X;"") = ";E:WAIT
555 CURSOR 0,2:PRINT "E(X) = var(x) = ";LAM
560 GOTO 500
570 "H":CLEAR
575 CLS:WAIT 1:PRINT "HYPERGEOM. VERTEILUNG"
580 INPUT "N = ";G
585 INPUT "M = ";M
590 INPUT "n = ";N
595 INPUT "x = ";X
600 Y=M:Z=X:GOSUB 650
605 E2=E1
610 Y=G-M:Z=N-X:GOSUB 650
615 E3=E1
620 Y=G:Z=N:GOSUB 650
625 E4=E1:GOTO 700
650 E1=1:F=1:IF Z=0 THEN 670
655 FOR A=(Y-Z+1)TO Y
660 E1=E1*A
665 NEXT A
670 FOR A=1 TO Z
675 F=F*A
680 NEXT A
685 E1=E1/F:RETURN
700 LET E1=INT(E2*E3*10^5/E4)/10^5
705 CLS:PRINT "N= ";G;" M= ";M;" n= ";N
715 PRINT "P(X=";X;"") = ";E1
720 ERW=INT(N*M*10^5/G)/10^5
725 VAR=(N*M*G-N*M^2)*(G-N)/(G^3-G^2)
726 VAR=INT(VAR*10^5)/10^5
730 PRINT "E(X) = ";ERW:WAIT
735 PRINT "var(x) = ";VAR
740 GOTO 575
750 "M":CLS:WAIT 1:CLEAR
755 PRINT "MULTINOMINALVERTEILUNG"
760 INPUT "WIEVIELE ZV ? ";R:DIM P(R):DIM X(R)
765 INPUT "n = ";N
770 FOR A=1 TO R
775 PRINT "p(";A;"") = ":INPUT P(A):LET P(0)=P(0)+P(A)
780 NEXT A
782 IF P(0)<>1 THEN 750
785 WAIT 1:E1=1:E2=1:F=1:X(0)=0
790 FOR A=1 TO R
795 PRINT "x(";A;"") = ":INPUT X(A):LET X(0)=X(0)+X(A)
800 NEXT A
803 IF X(0)<>N THEN 785
805 FOR A=1 TO N
810 E1=E1*A
815 NEXT A
820 FOR A=1 TO R
825 E2=E2*P(A)^X(A)
830 NEXT A
835 E1=E1*E2:E2=1
840 FOR B=1 TO R
842 F=1
845 FOR A=1 TO X(B)
850 F=F*A
855 NEXT A
860 E2=E2*F

```

```

865 NEXT B
870 E=INT(E1*10^5/E2)/10^5:CLS:WAIT
875 PRINT "P(X=xi) = ";E
880 GOTO 785

```

Bruchapproximation

Beschreibung:

Nach Eingabe einer Zahl vom Typ Real wird zu dieser Zahl eine rationale Zahl mit ganzzahligem Zähler und Nenner approximiert.

Listing:

```

100 REM BRUCHAPPROXIMATION
110 INPUT "X = ? ";X
120 DIM Y(12)
130 K=8:EP=1E-6
130 FOR I=1 TO K
140 Y(I)=INT(X+EP)
150 IF ABS(Y(I)-X)<EP THEN "EXIT"
160 X=RCP(X-Y(I)):NEXT I
170 "EXIT"
180 Z=Y(I):N=1
190 FOR J=I-1 TO 1 STEP-1
200 H=Z:Z=Y(J)*Z+N:N=H
210 NEXT J
220 PAUSE "DIVIDEND:"
230 PRINT "X = ";Z
240 PAUSE "DIVISOR:"
250 PRINT "X = ";N
260 END

```

Gleichungen 2. , 3. und 4. Grades

Beschreibung:

Hiermit ist es möglich, Gleichungen des 2., 3. oder 4. Grades zu berechnen. Wichtig hierbei ist, daß nicht nur die reellen Lösungen sondern auch die komplexe Lösung z.B der Gleichung $X^2 + 1 = 0$ ausgegeben wird.

Listing:

```

100 REM GLEICHUNGEN BIS ZUM 4. GRAD
110 INPUT "GRAD ? ";N
120 IF N>4 OR N<2 GOTO 110
130 PAUSE "KOEFF. EINGABE:"
140 FOR I=5-N TO 5
150 PAUSE CHR$(64+I);" = ? "
160 INPUT A(I):NEXT I:XG$="X="
170 ON 5-N GOTO "VIER", "DREI", "ZWEI"
180 "VIER"
190 IF B=0 AND D=0 GOTO "BIQUAD"

```

```

200 GOSUB "CUBE"
210 J=8*Y+B*B-4*C
220 IF ABS(J)<5E-8 THEN J=0
230 H1=SQR(J):G=H1:H=(B*H1+J)/2
240 I=(H1+B)*Y-D:GOSUB "QUAD"
250 H1=-H1:H=(B*H1+J)/2:G=H1
260 I=(H1+B)*Y-D:GOSUB "QUAD"
270 END
280 "CUBE"
290 A1=-C/2:B1=B*D/4-E
300 C1=(E*(4*C-B*B)-D*D)/8
310 "CUBIC" P=-A1*A1/9+B1/3
320 Q=A1^3/27-A1*B1/6+C1/2
330 R=Q*Q+P^3
340 ON 2-SGN(R) GOTO 350,390,420
350 U=SQR(R)-Q:U=SGN(U)*CUR ABS U
360 V=-SQR(R)-Q:V=SGN(V)*CUR ABS V
370 Y=U+V-A1/3
380 RETURN
390 U=SGN(-Q)*CUR ABS Q
400 Y=2*U-A1/3
410 RETURN
420 X=-Q/SQR(-P^3)
430 ON SGN(X)+2 GOTO 440,460,450
440 PH=ACS(X)+PI:GOTO 470
450 PH=ACS(X):GOTO 470
460 PH=PI/2
470 Y=2*SQR(-P)*COS(PH/3)-A1/3
480 RETURN
490 "QUAD"
500 DI=H*H-4*G*I
510 IF G<>0 OR H<>0 OR I<>0 GOTO 530
520 PAUSE "4-FACHE LOESUNG:" :PRINT XG$:SQR(SQR(E)):END
530 IF DI<0 GOTO 570
540 PRINT XG$;(-H+SQR(DI))/(2*G)
550 PRINT XG$;(-H-SQR(DI))/(2*G)
560 RETURN
570 PRINT XG$;-H/(2*G);"+I*";SQR(-DI)/(2*G)
580 PRINT XG$;-H/(2*G);"-I*";SQR(-DI)/(2*G)
590 RETURN
600 "BIQUAD"
610 K=C*C-4*A*E
620 IF K<0 GOTO 760
630 Y1=(SQR(K)-C)/(2*A)
640 Y2=(-SQR(K)-C)/(2*A)
650 IF Y1<0 GOTO 680
660 PRINT XG$;SQR(Y1)
670 PRINT XG$;-SQR(Y1):GOTO 700
680 USING "##.##":PRINT XG$;"I*";SQR(-Y1)
690 PRINT XG$;"-I*";SQR(-Y1)
700 IF Y2<0 GOTO 730
710 PRINT XG$;SQR(Y2)
720 PRINT XG$;-SQR(Y2):GOTO 750
730 USING "##.##":PRINT XG$;"I*";SQR(-Y2)
740 PRINT XG$;"-I*";SQR(-Y2)
750 END
760 RE=-C/(2*A):IM=SQR(-K)/(2*A)
770 BT=SQR(RE*RE+IM*IM)
780 R1=SQR((BT+RE)/2):I1=SQR((BT-RE)/2) ^

```

```

790 BE=SQR(R1*R1+I1*I1)
800 R2=SQR((BE+R1)/2):I2=SQR((BE-R1)/2)
810 USING "##.##":PRINT XG$:R2;"-I*";I2
820 PRINT XG$:R2;"+I*";I2
830 PRINT XG$;-R2;"-I*";I2
840 PRINT XG$;-R2;"+I*";I2
850 END
860 "DREI"
870 A1=C/B:B1=D/B:C1=E/B:GOSUB "CUBIC"
880 ON 2-SGN(R) GOTO 890,920,950
890 X2=-(U+V)/2-A1/3:X3=(U-V)/2*SQR(3)
900 PRINT XG$:Y:USING "##.##":PRINT XG$:X2;"+I*";X3
910 PRINT XG$:X2;"-I*";X3:END
920 X2=-U-A1/3
930 PRINT XG$:Y:PRINT XG$:X2
940 PRINT XG$:X2:END
950 PRINT XG$:Y
960 PRINT XG$;-2*SQR(-P)*COS((PH+PI)/3)-A1/3
970 PRINT XG$;-2*SQR(-P)*COS((PH-PI)/3)-A1/3:END
980 "ZWEI"
990 G=1/2:H=D/(2*C):I=E/(2*C):GOSUB "QUAD"
1000 END

```

Matrizen-Multiplikation

Beschreibung:

Hiermit werden zwei Matrizen multipliziert, die vom Grad $A*B$ und $B*C$ sein müssen, die Ergebnismatrix ist vom Typ $A*C$.

Listing:

```

500 REM MATRIX-MULTIPLIKATION
510 INPUT "A ZEILEN? ";Z
520 INPUT "A SPALTEN? ";S
530 INPUT "B SPALTEN? ";P
540 DIM A(Z,S),B(S,P)
550 FOR I=1 TO Z
560 FOR J=1 TO S
570 PAUSE "A(";I;J;"):"
580 INPUT "? ";A(I,J)
590 NEXT J:NEXT I
600 FOR I=1 TO S
610 FOR J=1 TO P
620 PAUSE "B(";I;J;"):"
630 INPUT "? ";B(I,J)
640 NEXT J:NEXT I
650 FOR J=1 TO Z
660 FOR K=1 TO P
670 T=0
680 FOR L=1 TO S
690 T=T+A(J,L)*B(L,K)
700 NEXT L
710 PRINT "C(";J;K;")= ";T
720 NEXT K:NEXT J
730 END

```

Matrizen-Inversion

Beschreibung:

Das Programm invertiert eine Matrix nach dem Algorithmus von Faddejew. Der Grad N der Matrix sollte nicht all zu groß sein, da die Laufzeit um die Ordnung $p(N)=N*(N^3+N^2+N)$ wächst. Als inverse Matrix bekommt man ihre ganzzahligen Komponenten geliefert, falls der DIVISOR ungleich 1 ist, so muß man jede Komponente durch den DIVISOR teilen. Anschließend wird die Determinante der Matrix sowie das charakteristische Polynom ausgegeben.

Listing:

```

100 REM MATRIX-INVERSION NACH FADDEJEW
110 INPUT "GRAD ? ";N
120 IF N>8 OR N<1 GOTO 110
130 DIM A(N,N),B(N,N),G(N,N),H(N,N),C(N)
140 FOR I=1 TO N
150 FOR J=1 TO N
160 PAUSE "A(";I;J;"): "
170 INPUT "? ";A(I,J)
180 B(I,J)=A(I,J)
190 NEXT J:NEXT I
200 S=4*(N/2-INT(N/2))-1
210 FOR I=1 TO N
220 P=0
230 FOR J=1 TO N
240 P=P+B(J,J)
250 NEXT J
260 C(I)=P/I
270 FOR J=1 TO N
280 B(J,J)=B(J,J)-C(I)
290 FOR K=1 TO N
300 H(J,K)=B(J,K)
310 NEXT K:NEXT J
320 FOR J=1 TO N
330 FOR K=1 TO N
340 IF I=N-1 LET G(J,K)=B(J,K)
350 B(J,K)=0
360 FOR L=1 TO N
370 (J,K)=B(J,K)+A(J,L)*H(L,K)
380 NEXT L:NEXT K:NEXT J
390 C(I)=C(I)*S
400 NEXT I
410 F=C(N)*S
420 IF F=0 THEN PRINT "MATRIX SINGULAER":GOTO 490
430 PAUSE "INVERSE MATRIX"
440 FOR I=1 TO N
450 FOR J=1 TO N
460 PRINT "A(";I;J;")=";G(I,J)*SGN F
470 NEXT J:NEXT I
480 PRINT "DIVISOR=";ABS F
490 PRINT "DET=";F*S
500 PAUSE "POLYNOM:"
510 FOR I=0 TO N-1
520 PRINT "C(";I;")=";C(N-I)
530 NEXT I
540 PRINT "C(";N;")=";-S
550 END

```

Matrizen-Orthogonalisierung

Beschreibung:

Man kann damit eine Matrix nach dem Gram-Schmidt-Verfahren orthogonalisieren. Nach Eingabe einer Matrix, deren Spaltenvektoren eine Basis bilden, wird die orthogonalisierte Matrix ausgegeben, deren Spaltenvektoren orthogonal zueinander stehen und deren Norm=1 ist.

Listing:

```

100 REM ORTHOGONALISIEREN EINER MATRIX
110 REM NACH DEM GRAM-SCHMIDT-VERFAHREN
120 INPUT "GRAD ? ";N
130 IF N<2 OR N>8 THEN 120
140 DIM A(N,N),B(N,N),MY(N,N),QN(N)
150 PAUSE "KOEFF.-EINGABE:"
160 FOR I=1 TO N
170 FOR J=1 TO N
180 PAUSE "A(";I;J;") ="
190 INPUT A(I,J)
200 NEXT J:NEXT I
210 PAUSE "ORTHOGONALMATRIX"
220 FOR I=1 TO N
230 FOR J=1 TO N
240 B(I,J)=A(I,J):NEXT J
250 J=1
260 IF J+1>I THEN 360
270 MY(I,J)=0
280 FOR K=1 TO N
290 MY(I,J)=MY(I,J)+A(I,K)*B(J,K)
300 NEXT K
310 MY(I,J)=MY(I,J)/QN(J)
320 FOR K=1 TO N
330 B(I,K)=B(I,K)-MY(I,J)*B(J,K)
340 NEXT K
350 J=J+1:GOTO 260
360 QN(I)=0
370 FOR J=1 TO N
380 QN(I)=QN(I)+B(I,J)*B(I,J)
390 NEXT J
400 FOR J=1 TO N
410 PRINT "C(";I;J;")= ";B(I,J)/SQR QN(I)
420 NEXT J:NEXT I
430 END

```

Integral

Beschreibung:

Mit diesem Programm kann ein bestimmtes Integral einer innerhalb des Intervalls stetigen Funktion berechnet werden. Die Berechnung erfolgt mittels der Simpsonschen Regel (siehe Bronstein, Taschenbuch der Mathematik) wobei die Genauigkeit des Ergebnisses von der Anzahl der Rechendurchläufe abhängt. Die Funktion des Integrals muß vorher in Zeile 2420 als $y=f(x)$ eingegeben werden, dann kann das Programm gestartet werden. Auf die Frage 'WIE OFT:' muß die Zahl der gewünschten Rechendurchläufe angegeben werden. Je größer diese Zahl ist, um so höher ist die Genauigkeit des Ergebnisses, aber auch die Rechenzeit des Programmes steigt! Erfahrungsgemäß ist bei zehn Durchläufen das Ergebnis auf 4 Stellen hinter dem Komma gesichert.

Listing:

```
2200 CLEAR
2210 INPUT "UNTERE GRENZE: ";A
2220 INPUT "OBERE GRENZE: ";B
2230 "A" S=0
2240 INPUT "WIE OFT : ";N
2250 X=A: GOSUB 2420: S=Y
2280 M=2*N: H=(B-A)/M
2300 FOR J=1 TO M-1
2310 X=A+J*H
2320 GOSUB 2420
2330 S=S+2*Y*(1+(J<>2*INT(.5*J)))
2340 NEXT J
2350 X=B: GOSUB 2420: S=S+Y
2380 I=S*H/3: BEEP 1
2400 "L" PRINT I
2410 END
2420 Y= COS X* SIN X
2430 RETURN
```

Gauss'scher Algorithmus

Beschreibung:

Das nachfolgende Programm formt eine Matrix nach den Regeln des Gauss'schen Algorithmus so um, daß das entsprechende Gleichungssystem gelöst werden kann. Nach dem Starten des Programms werden Sie nach der Anzahl der Zeilen und Spalten gefragt. Sind diese eingegeben erscheint "ZW-ERG.? 1/0", wobei eine "1" eingegeben wird, wenn Zwischenergebnisse ausgegeben werden sollen. Nun beginnt die Eingabe der Koeffizienten, welche anschließend zur Kontrolle nochmals ausgegeben werden. Nach etwas Rechenzeit zeigt Ihnen der Rechner die Zwischenergebnisse bzw. das Endergebnis an. Tritt in der Hauptdiagonalen eine 0 auf, so wird das Programm abgebrochen und es erscheint die Fehlermeldung "A(Z,Z)=0 !!!".

Listing:

```

10 "G" CLEAR
20 PAUSE "GAUSS ALGORITHM."
30 INPUT "ZEILEN=";M
40 INPUT "SPALTEN=";N
50 DIM A(M,N)
60 INPUT "ZW-ERG.? 1/0 ";A$
70 PRINT "KOEFFIZ.EINGABE"
80 FOR S=1 TO M
90 FOR T=1 TO N
100 WAIT 40
110 PRINT "A(";S;",";T;")=": INPUT A(S,T)
120 NEXT T
130 NEXT S
140 WAIT : GOSUB 490
150 IF R=1 THEN RETURN
160 MM=M
170 IF M>N THEN LET MM=N-1
180 FOR Z=1 TO MM
190 IF A(Z,Z)=0 THEN GOSUB 270: GOTO 190
200 B=0
210 IF A(Z,Z)<>1 THEN GOSUB 360
220 GOSUB 410
230 IF A$="1" THEN GOSUB 490
240 NEXT Z
250 IF A$="1" THEN END
260 GOSUB 490: END
270 REM = 0-ROUT
280 IF Z=M THEN GOTO 340
290 IF Z+B+1>M THEN GOTO 340
300 FOR X=1 TO N
310 C=A(Z,X):A(Z,X)=A(Z+B+1,X):A(Z+B+1,X)=C
320 NEXT X
330 B=B+1: RETURN
340 PRINT " A(Z,Z)=0 !!!": IF A$<>"1" THEN GOSUB 490
350 END
360 REM <> 1-ROUT
370 FOR X=N TO 1 STEP -1
380 A(Z,X)=A(Z,X)/A(Z,Z)
390 NEXT X
400 RETURN

```


Nullstellen-Berechnung

Beschreibung:

Diese Programm bestimmt die Nullstellen einer Funktion nach dem Vorzeichen-Wechselverfahren. Der Vorteil des Programmes gegenüber der REGULA FALSI liegt darin, daß das Programm kürzer ist und auch weniger Rechenzeit benötigt. Die zu bestimmende Funktion muß stetig sein, sonst wird das Programm vorzeitig abgebrochen! Vor dem Start des Programms muß dabei die Funktion, deren Nullstelle ermittelt werden soll in Zeile 20 eingegeben werden. Wird mit trigonometrischen Funktionen gerechnet, so muß vorher auf die Betriebsart RADIAN umgeschaltet werden. In Zeile 20 ist als Beispiel der Sinus einprogrammiert.

Listing:

```

10 GOTO 30
20 RADIAN :Y= SIN X
25 RETURN
30 INPUT "UNTERE GRENZE ";UG: INPUT "OBERE GRENZE ";OG
40 S=( ABS (UG)+ ABS (OG))/50:S1=S
50 X=UG: GOSUB 20:Y1=Y
55 IF X>=OG THEN PRINT " ENDE !": GOTO 30
60 X=X+S: GOSUB 20: IF SGN Y1= SGN Y THEN LET Y1=Y:
   GOTO 55
70 IF ABS (Y)<1E-7 THEN BEEP 1: USING "&&&#.*****^":
   PRINT "NS: ";X: USING :UG=X+S:S=S1: GOTO 50
80 X=X-S:S=S/2: GOTO 55
410 REM ALGORITM.
420 FOR Y=1 TO M
430 IF Y=Z THEN GOTO 470
440 FOR X=N TO 1 STEP -1
450 A(Y,X)=A(Y,X)+(-A(Y,Z)*A(Z,X))
460 NEXT X
470 NEXT Y
480 RETURN
490 FOR D=1 TO M
500 FOR E=1 TO N
510 Q= INT (A(D,E)*10000+0.5)/10000
520 PRINT "A(";D;",";E;")";Q
530 NEXT E
540 NEXT D
550 RETURN

```

Differentiation

Beschreibung:

Dieses Programm dient zur Berechnung des numerischen Wertes der ersten und zweiten Ableitung einer Funktion $y=f(x)$ an der Stelle x . Dies ist für die meisten technisch - wissenschaftlichen Anwendungen ausreichend und der Speicherplatzbedarf des Programmes bleibt in Grenzen. Das Programm läßt sich auch gut als Unterprogramm in eigene Programme einbinden. Das Ergebnis liegt dann jeweils in der Variable Z vor. Bevor das Programm gestartet wird, muß im PRO-Mode die Stammfunktion $y=f(x)$ in Zeile 1000 eingegeben werden. Dann kann mit RUN gestartet werden.

Listing:

```

10 "D": INPUT "X=";B: GOTO 15
11 GOTO 10
15 PRINT "X= ";B
20 H= TEN -3:X=B+H: GOSUB 1000
30 Z=Y:X=B-H: GOSUB 1000
40 Z=(Z-Y)/2/H: PRINT "F1(X) : ";Z
50 X=B+2*H: GOSUB 1000
60 Z=Y:X=B-2*H: GOSUB 1000
70 Z=Z+Y:X=B: GOSUB 1000
80 Z=(Z-2*Y)/4/H/H: PRINT "F2(X) : ";Z
90 END
1000 Y=F(X)
1010 RETURN

```

CholeskyBeschreibung:

Dieses Programm zerlegt eine quadratische Matrix A in die obere Dreiecksmatrix R, die mit ihrer transponierten Matrix multipliziert wieder die Matrix A ergibt: $R^t * R = A$

Zuerst werden Sie nach der Anzahl der Zeilen bzw. Spalten (quadratisch) gefragt, welche mit N bezeichnet wird. Danach folgt die Abfrage nach den Koeffizienten. Sind diese eingegeben, werden sie nochmals komplett im Display ausgedruckt, um die Eingabe nochmals zu überprüfen.

Nach dieser Ausgabe berechnet der Rechner die obere Dreiecksmatrix R und gibt diese ebenfalls aus. Danach wird die transponierte Matrix R^t (untere Dreiecksmatrix) ausgegeben.

Beispiel:	Anzeige:	Eingabe:
		RUN
	CHOLESKY N<6	
	N=	2
	KOEFFIZ.EINGABE	
	A(1,1)= ?	1
	A(1,2)= ?	2
	A(2,1)= ?	3
	A(2,2)= ?	4
	A(1,1) 1	«RETURN»
	A(1,2) 2	«RETURN»
	A(2,1) 3	«RETURN»
	A(2,2) 4	«RETURN»
	R(1,1) 1	«RETURN»
	R(1,2) 2	«RETURN»
	R(2,1) 0	«RETURN»
	R(2,2) 0	«RETURN»
	RT(1,1) 1	«RETURN»
	RT(1,2) 0	«RETURN»
	RT(2,1) 2	«RETURN»
	RT(2,2) 0	«RETURN»

Listing:

```

10 CLEAR :R=1: PRINT "CHOLESKY N<6"
20 INPUT "N=";N:M=N: DIM A(N,N): DIM R(N,N): GOSUB 260
30 R(1,1)= SQR A(1,1): FOR V=2 TO N:R(1,V)=A(1,V)/R(1,1):
  NEXT V
40 R(2,2)= SQR (A(2,2)- SQR R(1,2))
50 IF N<3 THEN 140
60 FOR V=3 TO N:R(2,V)=(A(2,V)-R(1,2)*R(1,V))/R(2,2):
  NEXT V
70 R(3,3)= SQR (A(3,3)- SQR R(1,3)- SQR R(2,3))
80 IF N<4 THEN 140
90 FOR V=4 TO N:R(3,V)=(A(3,V)-R(1,3)*R(1,V)-R(2,3)*
  R(2,V))/R(3,3): NEXT V
100 R(4,4)= SQR (A(4,4)- SQR R(1,4)- SQR R(2,4)-
  SQR R(3,4))
110 IF N<5 THEN 140
120 R(4,5)=(A(4,5)-R(1,4)*R(1,5)-R(2,4)*R(2,5)-R(3,4)*R
  (3,5))/R(4,4)
130 R(5,5)= SQR (A(5,5)- SQR R(1,5)- SQR R(2,5)- SQR
  R(3,5)- SQR R(4,5))
140 FOR D=1 TO M
150 FOR E=1 TO N
160 Q= INT (R(D,E)*10000+0.5)/10000
170 PRINT "R(";D;",";E;")";Q
180 NEXT E
190 NEXT D
200 FOR D=1 TO M
210 FOR E=1 TO N
220 Q= INT (R(E,D)*10000+0.5)/10000
230 PRINT "RT(";D;",";E;")";Q
240 NEXT E
250 NEXT D
260 PRINT "KOEFFIZ.EINGABE"
270 FOR S=1 TO M
280 FOR T=1 TO N
290 WAIT 40
300 PRINT "A(";S;",";T;")=": INPUT A(S,T)
310 NEXT T
320 NEXT S
330 WAIT
340 FOR D=1 TO M
350 FOR E=1 TO N
360 Q= INT (A(D,E)*10000+0.5)/10000
370 PRINT "A(";D;",";E;")";Q
380 NEXT E
390 NEXT D
400 RETURN

```

Quadratische Gleichung

Beschreibung:

Das vorliegende Programm berechnet eine quadratische Gleichung der Form:

$$AX^2+BX+C=0$$

Das besondere dabei ist, daß nicht nur reelle Lösungen berechnet werden, sondern auch eventuell auftretende imaginäre Ergebnisse berechnet und angezeigt werden. Das Programm ist daher insbesondere für naturwissenschaftliche Berechnungen wie z.B. Elektrotechnik interessant.

Listing:

```

10 PAUSE "QUADRAT. GLEICH."
20 CLEAR : RADIANT
30 INPUT "A2 = ";A: INPUT "A1 = ";B: INPUT "A0 = ";C
40 IF A=0 THEN PRINT " FEHLER !!!": GOTO 10
50 D=B^2-(4*A*C)
60 IF D<0 THEN BEEP 1: PAUSE "KOMPLEXE LOESUNG": GOTO 80
70 GOTO 130
80 R=(B/A/2)*(-1):I= SQR (C/A- SQU (B/A/2))
90 IF ABS R<1 THEN USING "##.#####^"
100 PRINT "RE= ";R: USING
110 IF ABS I<1 THEN USING "##.#####^"
120 PRINT "IM= ";I: GOTO 10
130 X1=(-B+ SQR (D))/(2*A):X2=(-B- SQR (D))/(2*A)
140 IF ABS X1<1 THEN USING "##.#####^"
150 PRINT "X1=";X1: USING
160 IF ABS X2<1 THEN USING "##.#####^"
170 PRINT "X2=";X2
180 USING : GOTO 10

```

Größter gemeinsamer Teiler

Beschreibung:

Das Programm errechnet den größten gemeinsamen Teiler der beiden Zahlen A und B. Anschließend teilt es die Zahlen A und B durch diesen Teiler und zeigt das Ergebnis an. Dies erspart das nochmalige Eintippen der beiden Zahlen, denn meist schließt sich ja diese Operation an das Aufsuchen des größten gemeinsamen Teilers an.

Listing:

```

1000 REM ** G G T **
1010 PAUSE "GROESSTER GEM." : PAUSE "TEILER"
1020 INPUT "A=";A : INPUT "B=";B
1025 K=A : L=B
1030 Q=INT(A/B)
1040 R=A-(Q*B)
1050 A=B : B=R
1060 IF R>0 THEN 1030
1070 PRINT "GGT=";A
1080 K=K/A : L=L/A : PRINT "A/GGT=";K : PRINT "B/GGT=";L
1090 GOTO 1010

```

Kleinstes gemeinsames Vielfaches

Beschreibung:

Dieses kleine Programm ermittelt das kleinste, gemeinsame Vielfache der drei Zahlen A,B,C . Wer oft mit Brüchen rechnet ist sich über den Rechenaufwand bewußt, der normalerweise aufgebracht werden muß, um solche Zahlen zu berechnen. Das Programm, welches mit 'DEF X' gestartet wird, erledigt diese Arbeit.

Listing:

```

1200 "X"
1210 PAUSE "KLEINSTES GEM." : PAUSE "VIELFACHES"
1220 INPUT "A=";A : INPUT "B=";B : INPUT "C=";C
1230 X=A
1240 IF INT (X/A)=X/A THEN 1260
1250 X=X+1 : GOTO 1240
1260 IF INT (X/B)=X/B THEN 1280
1270 X=X+1 : GOTO 1240
1280 IF INT (X/C)=X/C THEN 1300
1290 X=X+1 : GOTE 1240
1300 PRINT "KGV=";X
1310 GOTO 1210

```

Regula Falsi (Nullstellen)

Beschreibung:

Dieses Programm ermittelt die Nullstellen einer Funktion, die vorher in Zeile 5300 eingegeben wurde. Das Programm sucht die Nullstelle zwischen 2 Intervallgrenzen X1 und X2. Dabei muß X1 kleiner X2 sein. Das Programm schiebt den X3 Punkt solange an die echte Nullstelle heran, bis das Ergebnis auf 6 Stellen hinter dem Komma gesichert ist. Diese Genauigkeit wird jedoch mit einer relativ großen Rechenzeit (bis 1 min) erkaufte. WICHTIG: Bei trigonometrischen Funktionen muß vorher auf RADIAN umgeschaltet werden. Dies kann manuell oder durch eine Programmzeile erfolgen.

Listing:

```

5000 "A"
5010 PAUSE "REGULA FALSI" : PAUSE "FU(X) IN Z. 5300" :
      PAUSE "NULLST. ZWISCHEN"
5020 INPUT "X1="; C : INPUT "X2="; D
5030 S=(D-C)/10
5040 GOSUB 5090
5045 H=Y : G=X
5050 C=C+S : GOSUB 5090
5060 IF H<0 AND Y>0 THEN 5150
5070 IF H>0 AND Y<0 THEN 5220
5080 GOTO 5045
5090 X=C : IF X>D THEN 5120
5095 GOSUB 5300
5100 IF Y=0 THEN 5140
5110 RETURN

```

```

5120 BEEP (2) : PRINT "KEINE NULLSTELLE"
5130 GOTO 5010
5140 X3=X : GOTO 5250
5150 J=Y : GOSUB 5190
5155 G=X3 : H=Y : GOSUB 5190
5160 P=G-X3
5170 IF P>0.0000001 THEN 5155
5180 GOTO 5250
5190 X3=G-(((C-G)*H)/(J-H))
5200 X=X3 : GOSUB 5300
5210 RETURN
5220 J=Y : GOSUB 5190
5225 C=X3 : J=Y : GOSUB 5190
5230 P=C-X3
5240 IF P>0.0000001 THEN 5225
5250 BEEP (1) : PAUSE "ERGEBNIS"
5260 USING "&&&##.#####^"
5270 PRINT "X3=" ; X3 : PRINT "Y3=" ; Y : USING
5280 GOTO 5010
5300 Y=2*X^3-6*X+1
5310 RETURN

```

Vektorberechnungen

Beschreibung:

Im Einzelnen lassen sich folgende Vektoroperationen durchführen:

Berechnung von:

- Skalarprodukt
- Kreuzprodukt
- Spatprodukt
- Betrag von Vektoren
- Winkel zwischen Vektoren

Außerdem gibt es noch ein Unterprogramm zur Berechnung von:

- Quadratischen Gleichungen

Erläuterungen im Einzelnen

1. Ein Vektor wird üblicherweise in Koordinatenschreibweise angegeben:
z.B. Vektor $A = (A_x / A_y / A_z)$ im 3D-Raum R^3
auch in diesem Programm werden alle Vektoren so eingegeben.

2. Def- Belegungen:

Nach der Betätigung von:

- » Def « » , « erscheint das Programmü
- » Def « » S « startet das Unterprogramm " Spatprodukt "
- » Def « » K « startet das Unterprogramm " Kreuzprodukt "
- » Def « » L « startet das Unterprogramm " Skalarprodukt "
- » Def « » B « startet das Unterprogramm " Betrag eines Vektors "
- » Def « » D « startet das Unterprogramm " Winkel zwischen Vektoren "
- » Def « » M « startet das Unterprogramm " Quadratische Gleichung "

3. Beschreibung der einzelnen Unterprogramme:

Skalarprodukt: Eingegeben werden jeweils nacheinander die Koordinaten zweier Vektoren A und B (Ax;Ay;Az) bzw. (Bx;By;Bz)
Ausgegeben wird der Wert des Skalarprodukts von A und B.

Spatprodukt: Eingegeben werden - siehe Skalarprodukt - diesmal drei Vektoren A, B und C . Der Rechner berechnet das Spatprodukt (Volumen des von diesen 3 Vektoren im Raum gebildeten Spats).

Kreuzprodukt: Es werden zwei Vektoren A und B in oben geschilderter Weise eingegeben, berechnet wird das Kreuzprodukt von A und B = A * B = A * B * sin (Winkel A,B)

Winkel: Es wird der Winkel zwischen den beiden eingegebenen Vektoren A und B berechnet. $\cos (\text{Winkel A,B}) = (A * B) : (/ A / * / B /)$

Betrag: Es wird der Betrag (d.h. die Länge) eines zuvor eingegebenen Vektors A berechnet. $/ A / = \text{SQR} (Ax * Ax + Ay * Ay + Az * Az)$

Quadratische Gleichung (oder auch " Mitternachtsformel " genannt):

Die Gleichung hat bekannterweise folgende Bauform:

$$AX^2+BX+C=0$$

Der Rechner fragt nun nacheinander die Glieder A, B und C ab und gibt anschliessend zwei, eine oder - gegebenenfalls - gar keine Lösung aus. Falls zwei Lösungen vorhanden sind, zeigt er zunächst die erste und nach Betätigung von » ENTER « die zweite Lösung an.

Listing:

```

1100
1150 "M"
1154 CLEAR:WAIT 50:PRINT " QUADR. GLEICHUNG"
1155 INPUT " A =";A
1160 INPUT " B =";B
1170 INPUT " C =";C
1175 D=(B*B-4*A*C):WAIT
1180 IF D=0 THEN 1210
1185 IF D<0 THEN 1212
1190 X=(-B+SQR(D))/2*A
1195 Y=(-B-SQR(D))/2*A
1200 PRINT "1. LOESUNG :";X:PRINT "2. LOESUNG :";Y:GOTO 1215
1210 PRINT "EINE LOESUNG : ";-B/(2*A):GOTO 1215
1212 PRINT " KEINE LOESUNG !"
1215 END
1220 "K"
1225 CLEAR
1230 WAIT 50:PRINT " KREUZPRODUKT  A # B "
1235 PRINT " VEKTOR A (X,Y,Z)"
1240 INPUT " AX =";X1
1245 INPUT " AY =";Y1
1250 INPUT " AZ =";Z1
1255 PRINT " A : (" ;X1;" , " ;Y1;" , " ;Z1;" )"
1260 PRINT " VEKTOR B (X,Y,Z)"
1265 INPUT " BX =";X2
1270 INPUT " BY =";Y2
1275 INPUT " BZ =";Z2
1280 PRINT " B : (" ;X2;" , " ;Y2;" , " ;Z2;" )"
1285 I=(Y1*Z2)-(Z1*Y2)
1290 I=INT(I*1000+.5)/1000
1295 J=(X1*Z2)-(Z1*X2)
1300 J=0-J
1305 J=INT(J*1000+.5)/1000
1310 K=(X1*Y2)-(Y1*X2)
1315 K=INT(K*1000+.5)/1000
1320 WAIT:PRINT " KP = (+I/--J/+K)"
1325 PRINT " KP : (I = " ;I;" )"
1330 PRINT " KP : (J = " ;J;" )"
1335 PRINT " KP : (K = " ;K;" )"
1340 INPUT " C*( A#B)  C=" ;C
1345 D=C*I
1350 D=INT(D*1000+.5)/1000
1355 E=C*J
1360 E=INT(E*1000+.5)/1000
1365 F=C*K
1370 F=INT(F*1000+.5)/1000
1375 PRINT " C*I = (" ;D;" )"
1380 PRINT " C*J = (" ;E;" )"
1385 PRINT " C*K = (" ;F;" )"
1390 END
1395 "S"
1400 CLEAR:WAIT 50
1405 PRINT " SPATPRODUKT (A,B,C)"
1410 PRINT " VEKTOR A : (AX,AY,AZ)"
1420 INPUT " AX =";AX
1425 INPUT " AY =";AY
1430 INPUT " AZ =";AZ

```



```
1435 PRINT " VEKTOR B : (BX, BY, BZ)"
1440 INPUT " BX ="; BX
1445 INPUT " BY ="; BY
1450 INPUT " BZ ="; BZ
1455 PRINT " VEKTOR C : (CX, CY, CZ)"
1460 INPUT " CX ="; CX
1465 INPUT " CY ="; CY
1470 INPUT " CZ ="; CZ
1475 A=AX*(BY*CZ-BZ*CY)
1480 B=AY*(BX*CZ-BZ*CX)
1485 C=AZ*(BX*CY-BY*CX)
1490 PRINT " SP = ( X, Y, Z )"
1495 SP=A-B+C: IF SP<0 THEN SP=SP*(-1)
1500 WAIT:PRINT " SPATPRODUKT = "; SP
1501 END
1600 "L":CLEAR:WAIT 50
1605 PRINT " SKALARPRODUKT "
1610 PRINT " VEKTOR A * VEKTOR B"
1615 PRINT " VEKTOR A = (AX, AY, AZ)"
1620 INPUT " AX ="; AX
1625 INPUT " AY ="; AY
1630 INPUT " AZ ="; AZ
1635 PRINT " VEKTOR B = (BX, BY, BZ)"
1640 INPUT " BX ="; BX
1645 INPUT " BY ="; BY
1650 INPUT " BZ ="; BZ
1655 D=(AX*BX)+(AY*BY)+(AZ*BZ)
1660 WAIT:PRINT " SKALARPRODUKT = "; D
1665 END
1670 "B":CLEAR:WAIT 50
1675 PRINT " BETRAG EINES VEKTORS "
1680 PRINT " VEKTOR A = (AX, AY, AZ)"
1685 INPUT " AX ="; AX
1690 INPUT " AY ="; AY
1695 INPUT " AZ ="; AZ
1700 B=SQR(AX*AX+AY*AY+AZ*AZ)
1705 WAIT:PRINT "BETRAG VON A ="; B
1710 "D":CLEAR:WAIT 50
1715 PRINT " WINKEL ZWISCHEN VA UND VB"
1720 PRINT " VEKTOR A = (AX, AY, AZ)"
1725 INPUT " AX ="; AX
1730 INPUT " AY ="; AY
1735 INPUT " AZ ="; AZ
1740 PRINT " VEKTOR B = (BX, BY, BZ)"
1745 INPUT " BX ="; BX
1750 INPUT " BY ="; BY
1755 INPUT " BZ ="; BZ
1760 BB=SQR(BX*BX+BY*BY+BZ*BZ)
1765 AB=SQR(AX*AX+AY*AY+AZ*AZ)
1770 W=(AX*BX+AY*BY+AZ*BZ)/(BB*AB)
1775 WI=ACS W:WI=INT(WI*100+.5)/100
1780 WAIT:PRINT "WINKEL(A, B) ="; WI; " GR."
1786 END
7000 " "
7010 CLEAR
7012 WAIT 60:PRINT " PROGRAMMENUE"
7014 PRINT " S -> SPATPRODUKT"
7016 PRINT " K -> KREUZPRODUKT"
7018 PRINT " L -> SKALARPRODUKT"
```

```

7020 PRINT " B -> BETRAG V. VEKT."
7022 PRINT " D -> WINKEL <> VEKT."
7024 PRINT " M -> QUADR. GLEICHUNG "
7030 INPUT " WIEDERHOLUNG ? (J)";JS
7035 IF JS="J" THEN GOTO 7012
7040 END

```

Rechnen mit komplexen Zahlen und Gleichungssystemen

Beschreibung:

Dieses Programm ermöglicht das Rechnen mit komplexen Zahlen, sowie das lösen von komplexen Gleichungssystemen.

Beim Rechnen stehen folgende Funktionen zur Verfügung :

+ - * / ^ = ,Wurzel (SQR),Kehrwert (RCP),e hoch x,Polar «-» Rechtw.

Im Gegensatz zu älteren Versionen wurde auf trigonometrische und andere speziellen Funktionen verzichtet.

Bedienung:

Es kann nur mit Variablen gerechnet werden, welche vorher definiert werden müssen. Es stehen 7 Variablen zur Verfügung (A,S,D,F,G,H,J). Drücken Sie die DEF Taste und anschliessend eine der 7 Variablen. Nun können Sie den komplexen Wert eingeben.

Das Programm unterscheidet 2 verschiedene Ein- und Ausgabe Arten:

Polarmodus: die Betriebszustandsmarke 'STAT' leuchtet. Die Variableneingabe erfolgt nun polar (Betrag, Winkel). Alle Resultate werden polar ausgegeben.

Rechthöckig-Modus: 'STAT' leuchtet nicht. Ein- und Ausgabe erfolgen rechthöckig (X,Y).

Starten Sie das Programm mit RUN oder DEF K (Variablen bleiben erhalten).Nun können Sie den gewünschten Modus wählen (polar oder rechthöckig) indem Sie P (ENTER) oder R (ENTER) eingeben. Der Modus kann jederzeit gewechselt werden, je nach Bedarf. Jetzt müssen Sie Variablen definieren. Danach können Sie mit ihnen rechnen (Syntax wie folgt):

Beispiel: A= -3 + j5 S= 5.3 - 68.2 Grad (Polarform)

K:A+S	(ENTER)	-1.03175 7.9E-02 (d.h. -1.03175+j0.079)
A-S	(ENTER)	11.0954 116.600 (d.h. Betrag=11.0954 Winkel=116.600 Grad)
A/S		1.10017-170.836 (P)
A*S		30.9040 52.7637 (P)
A^S		-198630 1.0E 06 (R)
RCPS		7.0E-02 1.7E-01 (R)
SQRA		1.18973 2.10130 (R)
EA	(e hoch A)	4.9E-02-73.5211 (P)
D=R	das letzte Resultat wird der Variablen D zugewiesen.	

Das letzte Resultat wird immer in der Variablen R abgelegt. Mit R kann wie mit den anderen Variablen weiter gerechnet werden. Im weiteren werden auch die letzten 5 Resultate in einem Stack gespeichert. Man kann sie wie folgt zurückholen:

DEF SPC drücken. Nun erscheint das letzte Resultat. Drücken Sie nun mehrmals ENTER und die 5 letzten Resultate werden in einer Endlosschleife nacheinander angezeigt. Wenn nun der gewünschte Wert in der Anzeige steht, drücken Sie auf eine der 7 Variablen, in welche der Wert abgelegt werden soll. Nun noch DEF M. Das war's auch schon.

Mit DEF , können Sie jederzeit 'sauber' das Programm beenden.

Variabelinhalt anzeigen :

A (ENTER) zeigt den aktuellen Inhalt von A im aktuellen Modus an.
RR (ENTER) für das letzte Resultat.

Um einen Wert von der Polarform in die rechtwinklige Form umzuwandeln oder umgekehrt, gibt man den Wert ein, schaltet den Modus um und läßt sich den Wert wieder anzeigen.

Es kann in allen 3 Winkelmodi (DEG, RAD, GRAD) gerechnet werden.

Komplexe Gleichungssysteme

Starten mit DEF Z

ANZAHL GL.= (Geben Sie die Anzahl der Gleichungen Ihres Systems ein.)
SYMM.? (1/0): (Ist das Gleichungssystem symmetrisch zur Hauptdiagonalen? 1=Ja 0=Nein . Wenn Ja wird nur die Hälfte eingelesen, was bei großen Systemen Zeit spart.)
AUSG. R (1/0) (Soll die Ausgabe rechtwinklig erfolgen?)

Nun folgt die Eingabe der Matrix. Die Werte können **nur** rechtwinklig erfolgen. RE(1,1) heißt Realteil des Elementes der 1.Zeile und 1.Spalte. Sind alle Elemente eingegeben worden erfolgt die Berechnung.

Die Resultate werden in absteigender Reihenfolge mit U (in Anlehnung an die Elektrotechnik) bezeichnet und angezeigt. (z.B. U3=, U2=, U1=).

Listing:

```

10 REM KOMPLEX V3.0 17.5.87 BY D.LOERTSCHER
20 CLEAR: DIM RX(5), RY(5): RF=1: WAIT: DEGREE: GOTO "K"
45 AR=R: AI=I: RETURN
48 DR=R: DI=I: RETURN
50 FR=R: FI=I: RETURN
51 GR=R: GI=I: RETURN
52 HR=R: HI=I: RETURN
54 JR=R: JI=I: RETURN
62 RR=R: RI=I: RETURN
63 SR=R: SI=I: RETURN
65 R=AR: I=AI: RETURN
68 R=DR: I=DI: RETURN
70 R=FR: I=FI: RETURN

```

```

71 R=GR:I=GI:RETURN
72 R=HR:I=HI:RETURN
74 R=JR:I=JI:RETURN
82 R=RR:I=RI:RETURN
83 R=SR:I=SI:RETURN
85 R2=AR:I2=AI:RETURN
88 R2=DR:I2=DI:RETURN
90 R2=FR:I2=FI:RETURN
91 R2=GR:I2=GI:RETURN
92 R2=HR:I2=HI:RETURN
94 R2=JR:I2=JI:RETURN
102 R2=RR:I2=RI:RETURN
103 R2=SR:I2=SI:RETURN
104 "K" POKE 24636,8:IF RF POKE 24636,0
105 K$="":INPUT "K:";K$:L=LEN K$:IF L>4 THEN 1000
110 ON L GOTO 300,400,500,600
120 GOTO 105
130 "T"Z=0:IF(C$="A" OR C$="S" OR C$="D" OR C$="R" OR C$
="F" OR C$="G" OR C$="H" OR C$="J")THEN RETURN
140 Z=1:RETURN
160 "A" IF RF INPUT "RE=";AR,"IM=";AI:GOTO "K"
165 GOSUB 250:AR=Y:AI=Z:GOTO "K"
170 "S" IF RF INPUT "RE=";SR,"IM=";SI:GOTO "K"
175 GOSUB 250:SR=Y:SI=Z:GOTO "K"
176 "D" IF RF INPUT "RE=";DR,"IM=";DI:GOTO "K"
177 GOSUB 250:DR=Y:DI=Z:GOTO "K"
180 "F" IF RF INPUT "RE=";FR,"IM=";FI:GOTO "K"
185 GOSUB 250:FR=Y:FI=Z:GOTO "K"
190 "G" IF RF INPUT "RE=";GR,"IM=";GI:GOTO "K"
195 GOSUB 250:GR=Y:GI=Z:GOTO "K"
200 "H" IF RF INPUT "RE=";HR,"IM=";HI:GOTO "K"
205 GOSUB 250:HR=Y:HI=Z:GOTO "K"
210 "J" IF RF INPUT "RE=";JR,"IM=";JI:GOTO "K"
215 GOSUB 250:JR=Y:JI=Z:GOTO "K"
250 INPUT "BETRAG=";B,"WINKEL=";W:Y=REC(B,W):RETURN
300 IF K$="P" THEN LET RF=0:POKE 24636,8:GOTO "K"
310 IF K$="R" THEN LET RF=1:POKE 24636,0:GOTO "K"
315 C$=K$:GOSUB "T":IF Z THEN 1000
320 GOSUB ASC C$:IF RF=1 THEN 350
330 GOTO 380
350 PRINT R,I:GOTO "K"
380 IF I=0 AND R=0 THEN 350
381 Y=POL(R,I):PRINT Y,Z:GOTO "K"
400 IF LEFT$(K$,1)="E" LET C$=RIGHT$(K$,1):GOSUB "T"
402 IF K$="RR" LET C$="R":GOTO 320
405 IF Z=1 THEN 1000
410 GOSUB 900:GOSUB ASC C$:RR=EXP R*COS I:RI=EXP R*SIN I
:GOSUB 910:GOTO 800
500 C$=LEFT$(K$,1):GOSUB "T":IF Z THEN 1000
505 CC$=C$:C$=RIGHT$(K$,1):GOSUB "T":IF Z THEN 1000
510 V$=MID$(K$,2,1):IF(V$="+" OR V$="-" OR V$="*" OR V$
="/" OR V$="^")THEN 520
515 GOTO 1000
520 GOSUB ASC CC$:GOSUB(ASC C$+20):GOTO V$
550 "+"RR=R+R2:RI=I+I2:GOTO 800
560 "-"RR=R-R2:RI=I-I2:GOTO 800
565 "/"Q=R2*R2+I2*I2:RR=(R*R2+I*I2)/Q:RI=(R2*I-R*I2)/Q
:GOTO 800
570 "*"RR=R*R2-I*I2:RI=R*I2+R2*I:GOTO 800

```

```

575 "=" GOSUB ASC C$:GOSUB(ASC CC$-20):
    GOTO "K"
580 "^" GOSUB 900:Y=POL(R,I):M=R2*LN Y-I2*Z:
    N=I2*LN Y+R2*Z
585 RR=EXP M*COS N:RI=EXP M*SIN N:GOSUB 910:GOTO 800
600 C$=RIGHT$(K$,1):GOSUB "T":IF Z THEN 1000
610 CC$=LEFT$(K$,3):IF CC$="SQR" THEN 650
620 GOSUB ASC C$:RR=R/(R*R+I*I):RI=-I/(R*R+I*I):
    GOSUB 800
650 GOSUB ASC C$:GOSUB 900:Y=POL(R,I):RR=SQR Y*COS(Z/2)
660 RI=SQR Y*SIN(Z/2):GOSUB 910:GOTO 800
800 GOSUB 920:IF RF PRINT RR,RI:GOTO "K"
810 IF RI=0 AND RR=0 THEN 850
820 Y=POL(RR,RI):PRINT Y,Z:GOTO "K"
850 PRINT RR,RI:GOTO "K"
900 U=PEEK 24700:RADIAN:RETURN
910 POKE 24700,U:RETURN
920 RX(0)=RR:RY(0)=RI:FOR I=5 TO 1 STEP -1:RX(I)=RX(I-1):
    RY(I)=RY(I-1):NEXT I:RETURN
930 " I=1:IF RF=0 THEN 950
935 PRINT RX(I),RY(I):I=I+1:IF I>5 LET I=1:
    PRINT "*****"
940 GOTO 935
950 IF RX(I)=0 AND RY(I)=0 PRINT RX(I),RY(I):I=I+1:
    GOTO 953
951 GOTO 955
953 IF I>5 LET I=1:PRINT "*****"
954 GOTO 950
955 Y=POL(RX(I),RY(I)):PRINT Y,Z:I=I+1:GOTO 953
960 GOTO 950
990 "M" AREAD C$:R=RX(I):I=RY(I):GOSUB(ASC C$-20):
    GOTO "K"
1000 PRINT "FALSCH EINGABE":GOTO "K"
1090 "Z" CLEAR:POKE &45D0,&E2,&74,&84,&13,6,&0B,&24,&67
1091 POKE &45D8,&0D,&29,&04,&25,&84,&13,6,&0B
1092 POKE &45E0,&78,&C4,&17,&D1,&02,&58,&32,&37
1100 INPUT "ANZAHL GL. =":N:DIM R(N-1,N),I(N-1,N):P=1:
    INPUT "SYMM.? (1/0)":S
1102 INPUT "AUSG. R (1/0)":RF
1105 FOR I=0 TO N-1:M=S*I:FOR J=M TO N
1110 WAIT 0:PRINT "RE(";STR$(I+1);", ";STR$(J+1);")=":
    CALL &45D0;R(I,J)
1115 PRINT "IM(";STR$(I+1);", ";STR$(J+1);")=":
    CALL &45D0;I(I,J)
1120 IF S=1 AND J<>N LET R(J,I)=R(I,J):I(J,I)=I(I,J)
1122 NEXT J:NEXT I:WAIT
1125 FOR K=0 TO N-2:FOR I=K+1 TO N-1:C=R(K,K):D=I(K,K):
    A=R(I,K):B=I(I,K)
1130 M=C*C+D*D:C=C/M:D=-D/M:M=A*C-B*D:B=B*C+A*D:A=M
1135 FOR J=K+1 TO N:E=R(K,J):F=I(K,J):
    R(I,J)=R(I,J)-E*A+F*B:I(I,J)=I(I,J)-F*A-E*B
1140 NEXT J:NEXT I:NEXT K
1145 Y=N-1:A=R(Y,N):B=I(Y,N):C=R(Y,Y):D=I(Y,Y):Q=0
1150 I=N-1
1155 Y=R(I,N):Z=I(I,N):J=I
1160 J=J+1:IF J<N-.1 LET Y=Y-R(I,J):Z=Z-I(I,J):GOTO 1160
1165 U=R(I,I):V=I(I,I):S=U*U+V*V:R=(Y*U+Z*V)/S:
    Z=(Z*U-Y*V)/S:Y=R

```

```

1170 FOR L=0 TO I-1:E=R(L,I):F=I(L,I):R(L,I)=E*Y-F*Z:
      I(L,I)=E*Z+F*Y:NEXT L
1175 PRINT "U"+STR$(I+1):GOSUB 1185
1180 I=I-1:IF I>-.1 THEN 1155
1182 END
1185 IF RF PRINT Y,Z:RETURN
1190 IF Y=0 AND Z=0 THEN 1200
1195 Y=POL(Y,Z):PRINT Y,Z:RETURN
1200 PRINT Y,Z:RETURN
1300 ", " END

```

Finanzmathematik

Beschreibung:

Das Programm wurde auf dem SHARP PC-1403 getestet und entwickelt !

```

DEF A : SPAREN
DEF B : SPARRATE
DEF C : RENTE
DEF D : RENDITE
DEF G : DARLEHN
DEF H : HYPOTHEK

```

Sparen: Der Sparverlauf für eine Sparrate R, die N-mal pro Jahr über einen Zeitraum von J Jahren bei einem Jahreszinssatz Z gezahlt wird, soll dargestellt werden. Das Sparkapital K wird in N Schritten pro Jahr gemäß $K(\text{neu}) = (K + R) * (1 + Z / (100N))$ gebildet und das jeweilige Jahresergebnis angegeben.

Beispiel: Höhe der Sparrate: 250
 Wie oft pro Jahr : 4
 Zahl der Jahre : 8
 Zinssatz pro Jahr : 4.5

JAHR	KAPITAL
1.	1028.44
2.	2103.95
3.	3228.68
4.	4404.89
...	...
8.	9673.2

Sparrate: Es ist die Sparrate zu bestimmen, die N-mal pro Jahr über einen Zeitraum von J Jahren gezahlt werden muß, um ein gewünschtes Endkapital K zu erhalten. Die "Zinsentwicklung" wird unabhängig vom Endkapital K durch N-malige Durchführung des Grundschrilles $S(\text{neu}) = (S+1) * (1+Z/(100N))$ für ein Jahr und J-malige Wiederholung für J Jahre nachgebildet. Die Sparrate R ergibt sich dann zu $R=K/S$ und wird gerundet angegeben.

Beispiel: Durch Probe des Beispiels zu SPAREN.

Rente: Es soll ermittelt werden, welche Rente N-mal pro Jahr aus einem Kapital K anfällt, wenn ein Jahreszinssatz Z und eine Laufzeit T zugrunde gelegt wird.

Die N-mal pro Jahr zu zahlende Rente bestimmt sich zu

$$R = K \left(\frac{Z/(100N)}{(1+Z/(100N))^{NJ} - 1} + \frac{Z}{100N} \right)$$

K = Anfangskapital
Z = Jahreszinssatz
J = Zahl der Jahre

Beispiel: Kapital : 80000
Raten pro Jahr : 12
Wieviel Jahre : 15
Zinssatz : 6.2

Rente 12mal pro Jahr ==> 683.76 DM

Rendite: Es soll ermittelt werden, welche Jahresrendite ein Wertpapier mit dem Nennwert(Ausgabewert) N und der Restlaufzeit T einbringt, wenn es zum Kaufpreis K erworben ist. Die Durchschnittsrendite in Prozent/Jahr ist

$$R = \frac{N*Z}{K} + 100 \frac{N - K}{K*T}$$

Z = Jahreszinssatz (Dividende)
K = Kaufpreis
N = Nennwert
T = Restlaufzeit

Beispiel: Kaufpreis : 58000
Nennwert : 60000
Zinssatz% : 6.5
Restlaufzeit : 7

Rendite (%) : 7.216748768

Darlehen: Tilgungs- und Zinsverlauf für ein Darlehen K soll ermittelt werden. Gegeben sind die Fälligkeit R N-mal pro Jahr und der Jahreszinssatz Z. Die jährlichen Zinsen und die Restschuld sowie die Gesamtzinsen und die Gesamtlaufzeit sind anzugeben. Zu jedem Fälligkeitstermin I (N-mal pro Jahr) wird der Zinsanteil $K*Z/(100*N)$ dem jeweiligen Kapital K als eine Schuld zugerechnet, während die Rate R vom jeweiligen Kapital subtrahiert wird. Nach jeweils n Schritten werden die Zinsen, die Tilgung und die Restschuld ausgegeben. Ist das Kapital innerhalb eines Jahresabschnitts getilgt, so wird der Ablauf vorzeitig beendet.

Beispiel: Darlehnsbetrag : 10000
Annuität : 250
Zahlung pro Jahr : 12
Jahreszinssatz : 6.8

Jahr	Zinsen	Tilgung	Restschuld
1	606.31	2393.69	7606.31
2	438.37	2561.63	5044.68
3	258.65	2741.35	2303.32
4	69.18	2303.32	0.

Laufzeit: 3 Jahre 10 Monate
Summe aller Zinsen: 1372.5 DM

Hypothek: Für eine Hypothek K , die mit einem Zinssatz Z und einem Tilgungssatz T pro Jahr abzuzahlen ist, sind die jeweiligen Jahreszinsen, die Jahrestilgung und die jeweilige Restschuld zu bestimmen. Außerdem ist die gesamte Laufzeit und die Höhe der Gesamtzinsen zu ermitteln. Zunächst wird die Fälligkeit R , die N -mal pro Jahr zu begleichen ist, durch $R = K(Z+T)/(100N)$ ermittelt und angegeben. In N Schritten pro Jahr wird der Zins- und Tilgungsanteil an der Fälligkeit R bestimmt und als jeweiliges Jahresergebnis angegeben. Die Fälligkeit R wird nicht nur zur Ausgabe gerundet, sondern auch gerundet in der weiteren Rechnung verwendet, da die jeweilige Zahlung nur in Mark und Pfennig erfolgt.

Beispiel: Höhe der Hypothek : 50000
 Zinssatz% (p.a.) : 6.7
 Tilgungssatz% (p.a.) : 2
 Zahlungen pro Jahr 4: 12

12 mal zu zahlen: 362.5 DM

Jahr	Zinsen	Tilgung	Restkapital
1	3318.71	1031.29	48968.71
2	3247.45	1102.55	47866.17
...
23	0.1	17.89	0.

Zinssumme: 45717.98 DM
 Laufzeit : 23. Jahre

Listing:

```

10 "A":PRINT "SPAREN"
20 CLEAR:INPUT "Hoehe der Sparrate: ";R
30 INPUT "Wie oft pro Jahr: ";N
40 INPUT "Zahl der Jahre: ";J
50 INPUT "Zinssatz pro Jahr ";Z
60 PRINT " JAHR      KAPITAL"
70 Z=1+Z/N/100
80 K=0
90 FOR T=1 TO J
100 FOR I=1 TO N
110 K=(K+R)*Z
120 NEXT I
130 PRINT "Jahr: ";T;"      ";INT(100*K+0.5)/100
140 NEXT T
150 FRAGES=" "
160 INPUT "NOCHMAL J/N? ";FRAGES
170 IF FRAGES="J" THEN 20
180 IF FRAGES="N" THEN 190
190 END
210 "B":PRINT "SPARRATE"
220 CLEAR:INPUT "Wie oft pro Jahr? ";N
230 INPUT "Wieviel Jahre? ";J
240 INPUT "Zinssatz pro Jahr? ";Z
250 INPUT "Welches Endkapital? ";K
260 Z=1+Z/N/100
270 S=0
280 FOR T=1 TO J
290 FOR I=1 TO N
300 S=(S+1)*Z

```



```

310 NEXT I
320 NEXT T
330 R=K/S
340 PRINT "Sparrate ";N;"mal"
350 PRINT "pro Jahr:";INT(R*100+0.5)/100;"DM"
360 FRAGES=" "
370 INPUT "Nochmal J/N?";FRAGES
380 IF FRAGES="J" THEN 220
390 IF FRAGES="N" THEN 400
400 END
510 "C":PRINT "RENTE"
520 CLEAR:INPUT "Kapital: ";K
530 INPUT "Raten pro Jahr: ";N
540 INPUT "Wieviel Jahre: ";J
550 INPUT "Zinssatz: ";Z
560 Z=Z/N/100
570 R=K*(Z/((Z+1)^(N*J)-1)+Z)
580 PAUSE "Rente ";N;"mal pro Jahr ==>"
590 PRINT "=="> ";INT(R*100+0.5)/100;" DM"
600 FRAGES=" "
610 INPUT "Neue Berechnung J/N? ";FRAGES
620 IF FRAGES="J" THEN 520
630 IF FRAGES="N" THEN 640
640 END
710 "D":PRINT "RENDITE"
720 CLEAR:INPUT "Kaufpreis: ";K
730 INPUT "Nennwert: ";N
740 INPUT "Zinssatz% ";Z
750 INPUT "Restlaufzeit (J): ";T
760 PRINT "Rendite (%): ";N*Z/K+100*(N-K)/(K*T)
770 FRAGES=" "
780 INPUT "Neue Berechnung J/N? ";FRAGES
790 IF FRAGES="J" THEN 720
800 IF FRAGES="N" THEN 810
810 END
910 "G":PRINT "DARLEHN (Tilgungsplan)"
920 CLEAR:INPUT "Darlehnsbetrag? ";K
930 INPUT "Annuitaet? ";R
940 INPUT "Zahlungen pro Jahr? ";N
950 INPUT "Jahreszinssatz? ";Z
960 Z=Z/100/N
970 Y=0
980 S=0
990 TI=K
1000 X=0
1010 FOR I=1 TO N
1020 X=X+K*Z
1030 K=K+K*Z-R
1040 IF K>0 THEN 1070
1050 K=0
1060 GOTO 1080
1070 NEXT I
1080 Y=Y+1
1090 PRINT Y;"tes Jahr ==>"
1100 PRINT "Zinsen: ";INT((100*X)+0.5)/100;" DM"
1110 PRINT "Tilgung: ";INT(100*(TI-K)+0.5)/100;" DM"
1120 PRINT "Restschuld: ";INT(100*K+0.5)/100;" DM"
1130 S=S+X
1140 IF K>0 THEN 990
1150 PRINT "LAUFZEIT:"

```

```

1160 PRINT Y-1;" Jahre ";12*I/N;" Monate"
1170 PRINT "Sume aller Zinsen:"
1180 PRINT INT(100*S+0.5)/100;" DM"
1190 FRAGES=" "
1200 INPUT "Neue Berechnung? J/N";FRAGES
1210 IF FRAGES="J" THEN 920
1220 IF FRAGES="N" THEN 1230
1230 END
1310 "H":PRINT "HYPOTHEK"
1320 CLEAR:INPUT "Hoehe der Hypothek? ";K
1330 INPUT "Zinssatz% (p.a.)? ";Z
1340 INPUT "Tilgungssatz% (p.a.)? ";T
1350 INPUT "Zahlungen pro Jahr? ";N
1360 Z=Z/100/N
1370 Y=0
1380 S=0
1390 R=INT(100*K*(Z+T/100/N)+0.5)/100
1400 PAUSE N;"mal pro Jahr"
1410 PRINT "zu zahlen: ";R;" DM"
1420 TI=K
1430 X=0
1440 FOR I=1 TO N
1450 X=X+K*Z
1460 K=K+K*Z-R
1470 IF K>0 THEN 1500
1480 K=0
1490 GOTO 1510
1500 NEXT I
1510 Y=Y+1
1520 PRINT Y;"tes Jahr ==>>"
1530 PRINT "Zinsen: ";INT(100*X+0.5)/100;" DM"
1540 PRINT "Tilgung: ";INT(100*(TI-K)+0.5)/100;" DM"
1550 PRINT "Restkapital: ";INT(100*K+0.5)/100;" DM"
1560 S=S+X
1570 IF K>0 THEN 1420
1580 PRINT "Zinssumme: ";INT(100*S+0.5)/100;" DM"
1590 PRINT "Laufzeit: ";Y;" Jahre"
1600 FRAGES=" "
1610 INPUT "Neue Berechnung J/N?";FRAGES
1620 IF FRAGES="J" THEN 1320
1630 IF FRAGES="N" THEN 1640
1640 END

```

Einlagenrechnung

Beschreibung:

Das Programm "Einlagenrechnung" (Sparkassenrechnung) errechnet die Laufzeit, die Einlagen sowie das Endkapital eines Sparguthabens. Wahlweise bei vorschüssiger (1 eingeben) oder nachschüssiger (2 eingeben) Zahlung der Einlagen.

Das Programm wird mit "DEF X" gestartet.

1. Beispiel

Herr Mayer legt an jedem Monatsanfang DM 250.00 auf ein Sparkonto. Der Guthabenzins beträgt 6.5% jährlich. Wie hoch ist das angesparte Endkapital nach 4.5 Jahren?

Eingabe: Anfangskapital= 0 Endkapital= 0 Zins= 6.5/12 (monatlich) Zeiteinheiten= 54 (4.5 Jahre=54 Monate) Einlage= 250
VOR= 1 (vorschüßig)

Ausgabe: Endkapital= DM 15717.70

zu Beispiel 1:

Nach welcher Zeit hätte Herr Mayer DM 20 000.00 angespart?

Eingabe: Anfangskapital= 0 Endkapital= 20 000 Zins= 6.5/12
Zeiteinheiten = 0 Einlage = 250 VOR= 1

Ausgabe: Zeiteinheiten = 66.34 Monate (ca. 5 Jahre 7 Monate)

zu Beispiel 1:

Welche Höhe müßte die monatliche Sparrate des Herrn Mayer haben, um nach 4.5 Jahren auf ein Endkapital von DM 20 000.00 zu kommen?

Eingabe: Anfangskapital= 0 Endkapital= 20 000 Zins= 6.5/12
Zeiteinheiten = 54 Einlage = 0 Vor = 1

Ausgabe: Einlage(Rate)= DM 318.11

2. Beispiel

Herr Müller eröffnet ein Sparkonto und zahlt gleichzeitig DM 3600 ein. Der Guthabenzins beträgt 5.75 %. Zu diesem Anfangskapital zahlt er dann 4 Jahre lang am Ende eines jeden Quartals DM 450 ein. Wie hoch ist das Endkapital nach 4 Jahren?

Eingabe: Anfangskapital= 3600 Endkapital= 0 Zins= 5.75/4 (1/4-jährlich)
Zeiteinheiten = 16 (4 Jahre= 16 Quartale) Einlage= 450 NACH= 2
(nachsüßig)

Ausgabe: Endkapital= DM 12554.38

zu Beispiel 2:

Nach welcher Zeit hätte Herr Müller DM 15 000 angespart?

Eingabe: Anfangskapital= 3600 Endkapital= 15000 Zins = 5.75/4
Zeiteinheiten = 0 Einlage = 450 NACH= 2

Ausgabe: Zeiteinheiten= 19.80 Quartale (ca. 5 Jahre)

zu Beispiel 2:

Wie hoch müßte die vierteljährliche Rate des Herrn Müller sein, um nach 4 Jahren DM 15 000 angespart zu haben?

Eingabe: Anfangskapital= 3600 Endkapital= 15000 Zins = 5.75/4
Zeiteinheiten = 16 Einlage = 0 NACH= 2

Ausgabe: Einlage(Rate)= DM 587.03

Listing:

```

1480 "X" PRINT "EINLAGENRECHNUNG":CLEAR:
      USING "#####.##"
1485 REM **** BONFELD 1987 DAUBMANN ****
1490 INPUT "ANFANGSKAPITAL=";A,"ENDKAPITAL=";
      B,"ZINS %=";P
1492 INPUT "ZEITEINHEITEN=";N,"EINLAGE=";Z
1495 Q=1+P*.01
1497 IF Z=0 GOTO 1520
1500 IF N=0 GOTO 1535
1502 INPUT "VOR=1    NACH=2",P
1505 ON P GOTO 1510,1511
1510 B=A*Q^N+Z*Q*((Q^N-1)/(Q-1)):GOTO 1515
1511 B=A*Q^N+Z*((Q^N-1)/(Q-1))
1515 PAUSE "ENDKAPITAL KN="
1518 PRINT "KN=";B:END
1520 INPUT "VOR=1    NACH=2",P
1522 ON P GOTO 1524,1526
1524 Z=((B-A*Q^N)*(Q-1))/(Q^N-1)/Q:GOTO 1528
1526 Z=((B-A*Q^N)*(Q-1))/(Q^N-1)
1528 Z=Z+.001
1530 PAUSE "EINLAGE Z="
1533 PRINT "Z=";Z:END
1535 INPUT "VOR=1    NACH=2",P
1538 ON P GOTO 1540,1541
1540 N=LN((B*(Q-1)/Q+Z)/(A*(Q-1)/Q+Z))/LN Q:GOTO 1545
1541 N=LN((B*(Q-1)+Z)/(A*(Q-1)+Z))/LN Q
1545 N=N+.0001
1548 PAUSE "ZEITEINHEITEN N="
1550 PRINT "N=";N:END

```

Rentenrechnung

Beschreibung:

Das Programm "Rentenrechnung" (Tilgungsrechnung) errechnet die Laufzeit, die Renten (Tilgungsraten) sowie das Endkapital eines Guthabens. Wahlweise bei vorschüßiger (1 eingeben) oder nachschüßiger (2 eingeben) Zahlung der Renten (Raten).

Das Programm wird mit "DEF Z" gestartet.

1. Beispiel

Herr Schmidt besitzt Ersparnisse in Höhe von DM 84 000. Die Bank, bei der er das Geld angespart hat, zahlt ihm 6.25 % Zins jährlich. Von dem Guthaben will er seiner Tochter 4.5 Jahre lang monatlich (am Ersten) DM 620 zukommen lassen. Wie hoch sind seine Ersparnisse nach 4.5 Jahren?

Eingabe: Anfang kapital= 84000 Endkapital= 0 Zins= 6.25/12 (monatlich)
 Zeiteinheiten = 54 (4.5 Jahre=54 Monate) Rente= 620
 VOR= 1(vorschüßig)

Ausgabe: Endkapital= DM 72452.64

zu Beispiel 1:

Nach welcher Zeit hätte Herr Schmidt noch ein Guthaben in Höhe von DM 60 000?

Eingabe: Anfangskapital= 84000 Endkapital= 60 000 Zins= 6.25/12
 Zeiteinheiten = 0 Rente = 620 VOR= 1

Ausgabe: Zeiteinheiten= 99.06 Monate (nach ca. 8 Jahren und 3 Monaten)

zu Beispiel 1:

Welche Höhe müßte die monatliche Zahlung des Herrn Schmidt haben, damit sein Guthaben nach 4.5 Jahren noch DM 60 000 beträgt?

Eingabe: Anfangskapital= 84000 Endkapital= 60 000 Zins= 6.25/12
 Zeiteinheiten = 54 Rente = 0 Vor = 1

Ausgabe: Rente (Rate)= DM 819.25

2. Beispiel

Ein Bauherr nimmt einen Kredit in Höhe von DM 124 000 zu 7.25 % Zinsen auf. Die vierteljährliche Zins- u. Tilgungsrate (nachschiebig) soll DM 3200 betragen.

Nach welcher Zeit ist der gesamte Kredit getilgt?

Eingabe: Anfangskapital= 124000 Endkapital= 0 Zins= 7.25/4 (1/4-jährlich)
 Zeiteinheiten= 0 Rente (Tilgungsrate)= 3200
 NACH= 2 (nachschiebig)

Ausgabe: Zeiteinheiten= 67.46 (Quartale = ca. 17 Jahre)

zu Beispiel 2:

Welche Höhe müßte die vierteljährliche Zins u. Tilgungsrate haben, damit der Kredit nach 14 Jahren getilgt ist?

Eingabe: Anfangskapital= 124000 Endkapital= 0 Zins= 7.25/4
 Zeiteinheiten = 56 (14 Jahre = 56 Quartale)
 Rente (Tilgungsrate)= 0 NACH= 2

Ausgabe: Rente (Tilgungsrate)= DM 3543.33

3. Beispiel

Zum Kauf eines Autos nimmt der Käufer einen Kredit in Höhe von DM 12500 zu 4.99 % Jahreszins. Die Laufzeit des Kredites soll 48 Monate betragen. Welche Höhe hat die vorschüßige, monatliche Zins- u. Tilgungsrate?

Eingabe: Anfangskapital= 12500 Endkapital= 0 Zins= 4.99/12 (monatlich)
 Zeiteinheiten = 48 Rente (Tilgungsrate)= 0 VOR= 1

Ausgabe: Rente (Tilgungsrate)= DM 286.61

zu Beispiel 3:

Nach wieviel Monaten wäre der gesamte Kredit zurückgezahlt, wenn monatlich DM 300 getilgt würden?

Eingabe: Anfangskapital= 12500 Endkapital= 0 Zins= 4.99/12
 Zeiteinheiten = 0 Rente = 300 VOR= 1

Ausgabe: Zeiteinheiten= 45.64 Monate

Listing:

```

1400 "Z" PRINT "RENTENRECHNUNG":CLEAR:
      USING "#####.##"
1405 REM **** BONFELD 1987 DAUBMANN ****
1410 INPUT "ANFANGSKAPITAL=";A,"ENDKAPITAL=";
      B,"ZINS %=";P
1412 INPUT "ZEITEINHEITEN=";N,"RENTE=";R
1415 Q=1+P*.01
1417 IF R=0 GOTO 1440
1420 IF N=0 GOTO 1455
1425 INPUT "VOR=1    NACH=2",P
1427 ON P GOTO 1430,1431
1430 B=A*Q^N-R*Q*((Q^N-1)/(Q-1)):GOTO 1435
1431 B=A*Q^N-R*((Q^N-1)/(Q-1))
1435 PAUSE "ENDKAPITAL KN="
1438 PRINT "KN=";B:END
1440 INPUT "VOR=1    NACH=2",P
1442 ON P GOTO 1445,1446
1445 R=((A*Q^N-B)*(Q-1))/(Q^N-1)/Q:GOTO 1448
1446 R=((A*Q^N-B)*(Q-1))/(Q^N-1)
1448 R=R+.0001
1450 PAUSE "RENTE R="
1453 PRINT "R=";R:END
1455 INPUT "VOR=1    NACH=2",P
1457 ON P GOTO 1460,1461
1460 N=LN((R-B*(Q-1)/Q)/(R-A*(Q-1)/Q))/LN Q:GOTO 1463
1461 N=LN((R-B*(Q-1))/(R-A*(Q-1)))/LN Q
1463 N=N+0.0001
1465 PAUSE "ZEITEINHEITEN N="
1468 PRINT "N=";N:END

```

Dreiecksberechnung

Beschreibung:

Das Programm errechnet die 3 Seiten und 3 Winkel eines rechtwinkligen oder schiefwinkligen Dreieckes.

Sind von einem Dreieck 3 Werte bekannt, (ausgenommen 3 Winkelwerte) so sind die übrigen 3 Werte zu berechnen.

Die Werte für die Seiten werden in Dezimalzahlen eingegeben, die Werte für die Winkel in Sexagesimalzahlen (Grad,Minuten,Sekunden).

Die errechneten Ergebnisse werden in den gleichen Einheiten ausgegeben.

Der Programmstart erfolgt mit "Def C".

1. Beispiel

Bekannt sind die Seiten B=120.00 und C=60.00, sowie der Winkel WA=30.00'00"

Eingabe: Seite A=0 Seite B=120 Seite C=60 Winkel WA=30 Winkel WB=0
Winkel WC=0 (ENTER)

Ausgabe: Seite A=74.358 Seite B=120.000 Seite C=60.000 Winkel
WA=30.00'00.0" Winkel WB=126.12'21.7" Winkel WC=23.47'38.3"

Ist ein Dreieck zu berechnen, das einen stumpfen Winkel beinhaltet, (> 90 Grad) so ist darauf zu achten, daß der stumpfe Winkel auf den Winkel WB gelegt wird.

2. Beispiel

Bekannt sind die Seiten A=74.35882 und B=120.00, sowie der Winkel WA=30.00'00"

Eingabe : Seite A=74.35882 Seite B=120.0 Seite C=0 Winkel WA=30
Winkel WB=0 Winkel WC=0 (ENTER)

Nun erscheint auf dem Display "Stumpf=1 Spitz=2"

Hier muß angegeben werden, ob der Winkel WB spitz (<90 Grad) oder stumpf (>90Grad) ist.

Ist der Winkel WB größer als 90 Grad, so ist eine 1 einzugeben (ENTER). Ist der Winkel WB kleiner als 90 Grad, so ist eine 2 einzugeben (ENTER). Bei WB genau 90 Grad kann eine 1 oder 2 eigegeben werden.

Ausgabe :(WB>90): Seite A=74.358 B=120.000 C=60.000
Winkel WA=30.00'00.0" Winkel WB=126.12'21.7"
Winkel WC=23.47'38.8"

Ausgabe :(WB<90): Seite A=74.358 B=120.000 C=147.846
 Winkel WA=30.00'00.0" Winkel WB=53.47'38.3"
 Winkel WC=96.12'21.7"

Wegen den häufigen Zeilensprüngen (Goto, Gosub) wurden bei der Programm-
 erstellung keine Zeilennummern zur Programmverzweigung gewählt. Die Pro-
 grammverzweigung erfolgt mit Sprungmarken. Dies hat den großen Vorteil,
 daß eine eventuelle Änderung der Zeilennummern ohne großen Aufwand und
 somit auch ohne unnötige Fehlerquellen geschehen kann.

Listing:

```

1140 "C" PRINT "DREIECK":CLEAR:DEGREE:
      USING "#####.###"
1145 REM **** BONFELD 1987 DAUBMANN ****
1150 INPUT "SEITE A=";A,"SEITE B=";B,"SEITE C=";C
1155 INPUT "WINKEL A=";WA,"WINKEL B=";WB,"WINKEL C=";WC
1160 WA=DEG WA:WB=DEG WB:WC=DEG WC
1163 IF B>0 AND C>0 AND WA>0 GOSUB "AA":GOTO "AD"
1166 IF A>0 AND C>0 AND WB>0 GOSUB "AB":GOTO "AD"
1169 IF A>0 AND B>0 AND WC>0 GOSUB "AC":GOTO "AD"
1172 IF A>0 AND B>0 AND C>0 THEN "AD"
1175 IF B=0 OR WA=0 OR WB=0 THEN "AE"
1180 A=B*SIN WA/SIN WB:WC=180-WA-WB:GOSUB "AC":GOTO "AF"
1183 "AE":IF A=0 OR WB=0 OR WA=0 THEN "AG"
1185 B=A*SIN WB/SIN WA:WC=180-WA-WB:GOSUB "AC":GOTO "AF"
1188 "AG":IF A=0 OR WB=0 OR B=0 THEN "AH"
1190 WA=ASN(A*SIN WB/B):WC=180-WA-WB:GOSUB "AC":
      GOTO "AF"
1193 "AH":IF C=0 OR WA=0 OR WC=0 THEN "AI"
1195 A=C*SIN WA/SIN WC:WB=180-WA-WC:GOSUB "AB":GOTO "AF"
1198 "AI":IF A=0 OR WC=0 OR WA=0 THEN "AJ"
1200 C=A*SIN WC/SIN WA:WB=180-WA-WC:GOSUB "AB":GOTO "AF"
1203 "AJ":IF A=0 OR WC=0 OR C=0 THEN "AK"
1205 WA=ASN(A*SIN WC/C):WB=180-WA-WC:GOSUB "AB":
      GOTO "AF"
1208 "AK" IF C=0 OR WA=0 OR A=0 THEN "AL"
1210 WC=ASN(C*SIN WA/A):WB=180-WA-WC:GOSUB "AB":
      GOTO "AF"
1213 "AL":IF C=0 OR WB=0 OR WC=0 THEN "AM"
1215 B=C*SIN WB/SIN WC:WA=180-WB-WC:GOSUB "AA":GOTO "AF"
1218 "AM":IF B=0 OR WC=0 OR WB=0 THEN "AN"
1220 C=B*SIN WC/SIN WB:WA=180-WB-WC:GOSUB "AA":GOTO "AF"
1223 "AN":IF C=0 OR WB=0 OR B=0 THEN "AO"
1225 WC=ASN(C*SIN WB/B):WA=180-WB-WC:GOSUB "AA":
      GOTO "AF"
1230 "AO":INPUT "STUMPF=1 SPITZ=2",Z
1233 ON Z GOTO "AP","AQ"
1240 "AP":IF WA=0 OR B=0 OR A=0 THEN "AR"
1243 WB=ASN(SIN WA*B/A):WB=180-WB:WC=180-WA-WB:
      GOSUB "AC":GOTO "AF"
1245 "AR":WB=ASN(B*SIN WC/C):WB=180-WB:WA=180-WB-WC:
      GOSUB "AA":GOTO "AF"
1248 "AQ":IF WA=0 OR B=0 OR A=0 THEN "AS"
1250 WB=ASN(SIN WA*B/A):WC=180-WA-WB:GOSUB "AC":
      GOTO "AF"
1253 "AS":WB=ASN(B*SIN WC/C):WA=180-WB-WC:GOSUB "AA":
      GOTO "AF"

```



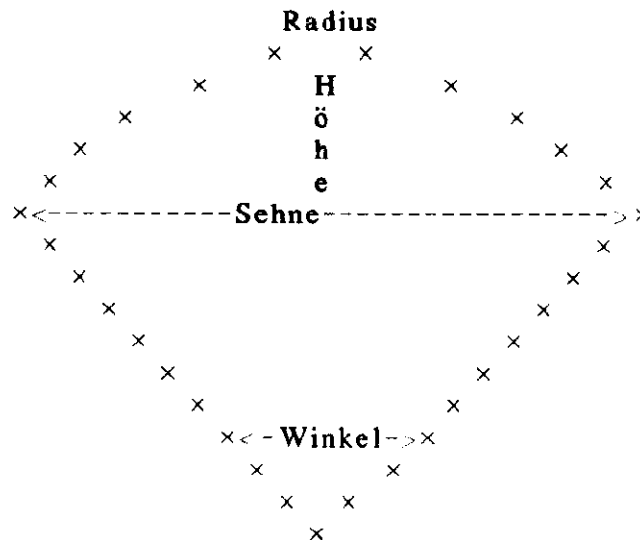
```
1260 "AD":IF WA>0 THEN "AT"
1263 WA=ACS((SQU B+SQU C-SQU A)/2/B/C)
1265 "AT":IF WB>0 THEN "AU"
1268 WB=ACS((SQU A+SQU C-SQU B)/2/A/C)
1270 "AU" IF WC>0 THEN "AF"
1273 WC=ACS((SQU A+SQU B-SQU C)/2/A/B)
1276 "AF":WA=WA+.000012:WB=WB+.00001:WC=WC+.00001
1278 WA=DMS WA:WB=DMS WB:WC=DMS WC
1280 PAUSE "SEITE A=":PRINT "A=";A:PAUSE "SEITE B=":
PRINT "B=";B:PAUSE "SEITE =C":PRINT "C=";C
1282 USING"****.*****"
1283 PAUSE "WINKEL WA=":PRINT "WA=";WA:
PAUSE "WINKEL WB=":PRINT "WB=";WB:PAUSE "WINKEL WC="
:PRINT "WC=";WC:GOTO "AV"
1285 "AA":A=SQR(SQU B+SQU C-2*B*C*COS WA):RETURN
1288 "AB":B=SQR(SQU A+SQU C-2*A*C*COS WB):RETURN
1290 "AC":C=SQR(SQU A+SQU B-2*A*B*COS WC):RETURN
1295 "AV":END
```

Kreisabschnitt

Beschreibung:

Das Programm errechnet die 4 Werte der abgebildeten geometrischen Figur.

a) Höhe b) Sehne c) Radius d) Winkel



Sind 2 Werte bekannt, so sind die übrigen 2 Werte zu berechnen.

Die Werte für Höhe, Sehne und Radius werden in Dezimalzahlen eingegeben, der Winkelwert in Sexagesimalzahlen (Grad, Minute, Sekunden).

Die errechneten Ergebnisse werden in den gleichen Einheiten ausgegeben.

Der Programmstart erfolgt mit "Def D".

Beispiel:

Bekannt ist die Höhe $H=12.35$ und der Winkel $W=35.12'46''$

Eingabe: $H=12.35$ $S=0$ $R=0$ $W=35.12'46''$

Ausgabe: $H=12.350$ $S=159.493$ $R=263.646$ $W=35.12'46.0''$

Listing:

```

1300 "D" PRINT "KREISABSCHNITT":CLEAR:DEGREE:
      USING "*****.***"
1305 REM *** BONFELD 1987 DAUBMANN ***
1307 INPUT "HOEHE=";H,"SEHNE=";S,"RADIUS=";
      R,"WINKEL=";W:W=DEG W
1310 IF R=0 OR W=0 OR H>0 THEN 1315
1312 H=R*(1-COS(W/2))
1315 IF S=0 OR W=0 OR H>0 THEN 1320
1317 H=S/2*(TAN(W/4))
1320 IF H=0 OR R=0 OR S>0 THEN 1325
1322 S=SQR(8*H*(R-H/2))

```

```

1325 IF H=0 OR S=0 OR R>0 THEN 1330
1327 R=H/2+SQU S/8/H
1330 IF S=0 OR R=0 OR W>0 THEN 1335
1332 W=ASN(S/2/R)*2
1335 IF H=0 OR W=0 OR S>0 THEN 1340
1337 S=2*H/TAN(W/4)
1340 IF H=0 OR S=0 OR R=0 OR W=0 THEN 1310
1342 W=DMS W
1345 PAUSE "HOEHE H=":PRINT "H=";H:PAUSE "SEHNE S=":
PRINT "S=";S
1348 PAUSE "RADIUS R=":PRINT "R=";R
1352 USING"####.####":PAUSE "WINKEL W=":PRINT "W=";W:
END

```

Bruchrechnen

Beschreibung:

1.) Werte eingeben:

Grundsätzlich wird der Bruch in folgender Form eingegeben:
 ganze Zahl/Zähler/Nenner

Das "/"-Zeichen dient dabei zur Trennung der Eingabewerte.

a) keine ganze Zahl

Ist in dem einzugebenden Bruch keine ganze Zahl enthalten, so kann diese einfach weggelassen werden.

b) kein Bruch

Soll nur eine ganze Zahl eingegeben werden, so kann diese direkt eingegeben werden.

Beispiele:

Wert	Eingabe
5 Ganze 3 Fünftel	5/3/5
2 Ganze	2
12 Sechsenddreißigstel	12/36

2.) Ausgabe

Nach jeder Eingabe erfolgt sofort die Umrechnung in den kleinsten Nenner und die Anzeige

a) Zähler und Nenner

b) ganze Zahl, Zähler und Nenner

c) Dezimalzahl

3.) Operationen

Es sind folgende Operationen mit Brüchen möglich

Operant		Taste
"+"	Addition	+
"-"	Subtraktion	-
"*"	Multiplikation	*
"/"	Division	/
"RCP"	Reziprokwert	1/x
"Q"	alles löschen	Q
"="	nochmal anzeigen	=
"U"	Umwandeln	U
"SQU"	Quadrat	x
"SQR"	Wurzel	$\sqrt{\quad}$

Hinweis zu der Funktion Umwandeln:

Ein Bruch kann in einen anderen Umgewandelt werden, dabei braucht der neue Nenner kein vielfaches des alten sein. Der Fehler der bei dieser möglichen Umwandlung entsteht wird angezeigt und die Durchführung abgefragt.

Beispiel:

2/3 umgewandelt 3/4 Fehler -1/12

Listing:

```

5000 "B":PAUSE "Bruchrechnen":CLEAR:GOTO "BSTART"
5010 "BEINGABE":G=0:Z=0:N=0:PAUSE "Bruch: G/Z/N":
      INPUT "";B$
5020 K=0:O=LEN(B$):FOR I=1 TO O
5030 IF MID$(B$,I,1)="/" THEN LET E=1:K=K+1
5040 IF E=1 AND K=1 THEN LET E=0:R=I:
      Z=VAL(LEFT$(B$,I-1)):N=VAL(RIGHT$(B$,O-I))
5050 IF E=1 AND K=2 THEN LET E=0:G=Z:
      Z=VAL(MID$(B$,R+1,O-I)):N=VAL(RIGHT$(B$,O-I))
5060 NEXT I
5070 IF K=0 THEN LET Z=VAL(B$)
5080 IF N=0 THEN LET N=1
5090 IF N<0 THEN LET N=ABS N:Z=-Z
5100 IF G<>0 THEN LET Z=G*N+Z
5110 Q1=Z:Q2=N
5120 RETURN
5130 "BAUSGABE":IF Z=0 THEN PRINT "0/0/0":GOTO 5210
5140 MINUS=0:IF Z<0 THEN LET MINUS=1:Z=ABS Z
5150 Q5=Z-INT(Z/N)*N:Q6=INT(Z/N)
5160 IF MINUS=1 THEN LET Z=-Z:Q6=-Q6
5170 IF Q5=0 THEN PRINT STR$ Q6;" / 0 / 0":GOTO 5210
5180 PRINT STR$ Z;" / ";STR$ N
5190 IF ABS Z>N THEN PRINT STR$ Q6;" / ";STR$ Q5;" / ";STR$ N
5200 PRINT Z/N
5210 RETURN
5220 "BSTART":GOSUB "BEINGABE"
5230 "BWEITER"
5240 IF Q2<>0 THEN IF Q1<>0 OR Q2<>0 THEN LET
      Q1=Q1-INT(Q1/Q2)*Q2:IF Q1>0 THEN LET
      Q2=Q2-INT(Q2/Q1)*Q1:GOTO 5240

```

```

5250 GT=Q1+Q2:Z=Z/GT:N=N/GT
5260 GOSUB "BAUSGABE":N1=N:Z1=Z
5270 BEEP 1:INPUT "Operation ";OP$:OP$="B"+OP$:GOTO OP$
5280 "B+":GOSUB "BEINGABE":Q1=Z1*N+N1*Z:Q2=N1*N:Z=Q1:
N=Q2:GOTO "BWEITER"
5290 "B-":GOSUB "BEINGABE":Q1=Z1*N-N1*Z:Q2=N1*N:Z=Q1:
N=Q2:GOTO "BWEITER"
5300 "B*":GOSUB "BEINGABE":Q1=Z1*Z:Q2=N1*N:Z=Q1:
N=Q2:GOTO "BWEITER"
5310 "B/":GOSUB "BEINGABE":Q1=Z1*N:Q2=N1*Z:Z=Q1:
N=Q2:GOTO "BWEITER"
5320 "B RCP ":Q1=N:Q2=Z:Z=Q1:N=Q2:GOTO "BWEITER"
5330 "BQ":CLEAR:GOTO "BSTART"
5340 "B=" GOTO 5260
5350 "BU" INPUT "Nenner umwandeln in ";N1:
USING "##.##^":Z1=Z*N1/N:IF (Z1-INT Z1)>0.5
THEN LET Z1=Z1+1
5360 A$="":Z1=INT Z1:PRINT STR$ Z1;":":STR$ N1;"
Fehler";(Z*N1-Z1*N)/N/N1:USING
5370 PRINT STR$ Z1;":":STR$ N1;" Fehler ";
STR$(Z*N1-Z1*N);":":STR$(N*N1):
INPUT " umwandeln ? ";A$
5380 IF A$="J" THEN LET Z=Z1:N=N1:Q1=Z:Q2=N
5390 IF A$="N" THEN GOTO "BU"
5400 GOTO "BWEITER"
5410 "B SQU "Z=SQU Z:N=SQU N:Q1=Z:Q2=N:GOTO "BWEITER"
5420 "B SQR "Z=SQR Z:N=SQR N:Q1=Z:Q2=N:GOTO "BWEITER"

```

Arithmetische Folgen und Reihen

Beschreibung:

Variablen bei Arithmetische Folgen und Reihen:

A1 - Erstes Glied
An - n-tes Glied
n - Anzahl
D - Differenz
Sn - Summe aller Glieder bis n-tes Glied

Alle Kombinationen sind möglich. Es darf höchstens eine Variable fehlen b.z.w. es kann immer nur eine gesucht werden.

Formeln:

$S_n = A_1 + A_2 + A_3 + \dots + A_n$
 $S_n = n/2 * (A_1 + A_n)$
 $A_n = A_1 + (n-1) * D$
 $D = A_{(n-1)} - A_n$

Listing:

```

10000 "Z" CLEAR:PAUSE " ARITHMETISCHE ":
      PRINT "FOLGEN U.REIHEN!"
10030 CLEAR:INPUT "GESUCHT IST ?";F$
10050 INPUT "WAS FEHLT ?";R$
10060 IF R$="A1" LET B=1
10062 IF R$="AN" LET B=2
10064 IF R$="N" LET B=3
10066 IF R$="D" LET B=4
10068 IF R$="SN" LET B=5
10080 REM A1 GESUCHT
10090 IF F$="A1" ON B GOTO 10030,10350,10380,10450,10500
10130 REM AN GESUCHT
10140 IF F$="AN" ON B GOTO 10550,10030,10600,10670,10720
10180 REM N GESUCHT
10190 IF F$="N" ON B GOTO 10770,10840,10030,10910,10960
10230 REM D GESUCHT
10240 IF F$="D" ON B GOTO 11010,11060,11110,10030,11160
10280 REM SN GESUCHT
10290 IF F$="SN" THEN ON B GOTO 11210,11260,11310,11360,
      10030
10310 REM FALSCH EING.
10330 GOTO 10030
10340 REM RECHENROUTINE
10350 INPUT "N=";N,"D=";D,"SN=";SN
10360 X=((2*SN/N)-((N-1)*D))/2:GOTO 11430
10380 INPUT "AN=";AN,"D=";D,"SN=";SN
10410 Q=-(AN+D)*AN+2*SN*D:P=-D:GOSUB 11480:GOTO 10030
10450 INPUT "AN=";AN,"N=";N,"SN=";SN
10480 X=(2*SN)/N-AN:GOTO 11430
10500 INPUT "AN=";AN,"N=";N,"D=";D
10530 X=AN-(N-1)*D:GOTO 11430
10550 INPUT "N=";N,"D=";D,"SN=";SN
10580 X=(2*SN+(N*D-D)*N)/(2*N):GOTO 11430
10600 INPUT "A1=";A1,"D=";D,"SN=";SN
10630 P=D:Q=D*(-2*SN+A1)-A1^2:GOSUB 11480:GOTO 10030
10670 INPUT "A1=";A1,"N=";N,"SN=";SN
10700 X=(2*SN)/N-A:GOTO 11430
10720 INPUT "A1=";A1,"N=";N,"D=";D
10750 X=A1+(N-1)*D:GOTO 11430
10770 INPUT "AN=";AN,"D=";D,"SN=";SN
10800 P=-(D+2*AN)/D:Q=(2*SN)/D:GOSUB 11480:GOTO 10030
10840 INPUT "A1=";A1,"D=";D,"SN=";SN
10870 P=(2*A1-D)/D:Q=-(2*SN)/D:GOSUB 11480:GOTO 10030
10910 INPUT "A1=";A1,"AN=";AN,"SN=";SN
10940 X=(2*SN)/(A1+AN):GOTO 11430
10960 INPUT "A1=";A1,"AN=";AN,"D=";D
10990 X=(AN-A1)/D+1:GOTO 11430
11010 INPUT "AN=";AN,"N=";N,"SN=";SN
11040 X=(2*SN-2*AN*N)/(-N^2+N):GOTO 11430
11060 INPUT "A1=";A1,"N=";N,"SN=";SN
11090 X=(2*SN-2*A1*N)/(N^2-N):GOTO 11430
11110 INPUT "A1=";A1,"AN=";AN,"SN=";SN
11140 X=(AN^2-A1^2)/(2*SN-A1-AN):GOTO 11430
11160 INPUT "A1=";A1,"AN=";AN,"N=";N
11190 X=(AN-A1)/(N-1):GOTO 11430
11210 INPUT "AN=";AN,"N=";N,"D=";D

```

```

11240 X=(N/2)*(2*AN-(N-1)*D):GOTO 11430
11260 INPUT "A1=";A1,"N=";N,"D=";D
11290 X=(N/2)*(2*A1+(N-1)*D):GOTO 11430
11310 INPUT "A1=";A1,"AN=";AN,"D=";D
11340 X=((AN-A1+D)/(2*D))*(A1+AN):GOTO 11430
11360 INPUT "A1=";A1,"AN=";AN,"N=";N
11390 X=(N/2)*(A1+AN):GOTO 11430
11420 REM AUSGABE
11430 PAUSE "*** GES.WERT ! ***":WAIT:PRINT F$;"=";X
11450 GOTO 10030
11470 REM QUAD.GLEICHUNG
11480 X1=- (P/2)+SQR((P/2)^2-Q)
11490 X2=- (P/2)-SQR((P/2)^2-Q)
11510 PAUSE "*** GES.WERT ! ***":WAIT:PRINT F$;" , 1 =" ;X1
11515 PRINT F$;" , 2 =" ;X2
11530 RETURN

```

Geometrische Folgen und Reihen

Beschreibung:

Variablen bei Geometrischen Folgen und Reihen:

A1 - Erstes Glied
 An - n-tes Glied
 n - Anzahl
 Q - Differenz
 Sn - Summe aller Glieder bis n-tes Glied

Einige Kombinationen sind nicht zu berechnen:

Gesucht wird An: Q und n fehlen - Nicht Möglich !
 Gesucht wird A1: Q und n fehlen - Nicht Möglich !
 Gesucht wird Q : An und A1 fehlen - Nicht Möglich !

Formeln:

$$A_n = A_1 * q^{(n-1)}$$

$$S_n = A_1 * \left(\frac{(q^n) - 1}{(q-1)} \right) \text{ und } |q| > 1$$

$$S_n = A_1 * \left(\frac{1 - (q^n)}{(1-q)} \right) \text{ und } |q| < 1$$

Listing:

```

1100 "X":PAUSE "* GEOMETRISCHE *":
    PRINT "FOLGEN U.REIHEN!"
1110 INPUT "GESUCHT IST ?";F$
1120 INPUT "WAS FEHLT ?";G$
1130 IF G$="AN" LET B=1
1140 IF G$="SN" LET B=2
1150 IF G$="A1" LET B=3
1160 IF G$="Q" LET B=4
1170 IF G$="N" LET B=5
1180 IF F$="A1" ON B GOTO 1300,1350,1290,1290,1290
1190 IF F$="AN" ON B GOTO 1290,1500,1400,1290,2020
1200 IF F$="N" ON B GOTO 1600,1700,1550,1650,1290
1210 IF F$="Q" ON B GOTO 1290,1800,1290,1290,1750
1220 IF F$="SN" ON B GOTO 1850,1290,1900,1970,1950

```

```

1290 PRINT " KEINE LOESUNG !":GOTO 1110
1300 INPUT "Q=";Q,"SN=";SN,"N=";N
1310 A1=(SN*(Q-1))/(Q^N-1)
1320 PRINT "A1=";A1
1330 GOTO 1110
1350 INPUT "AN=";AN,"N=";N,"Q=";Q
1360 A1=AN/(Q^(N-1))
1370 PRINT "A1=";A1:GOTO 1110
1400 INPUT "Q=";Q,"N=";N,"SN=";SN
1410 AN=Q^(N-1)*(SN/(Q^N-1)*(Q-1))
1420 PRINT "AN=";AN:GOTO 1110
1500 INPUT "A1=";A1,"Q=";Q,"N=";N
1510 AN=A1*Q^(N-1)
1520 PRINT "AN=";AN:GOTO 1110
1550 INPUT "AN=";A,"Q=";Q,"SN=";SN
1560 N=(LOG AN+LOG Q-LOG (SN-SN*Q+AN*Q))/(LOG Q)
1570 PRINT "N=";N:GOTO 1110
1600 INPUT "A1=";A1,"SN=";SN,"Q=";Q
1610 N=LOG((SN/A1)*(Q-1)+1)/(LOG Q)
1620 GOTO 1570
1650 INPUT "SN=";SN,"AN=";AN,"A1=";A1
1660 Q=(SN-A1)/(SN-AN)
1670 GOSUB 2000:GOTO 1110
1700 INPUT "AN=";AN,"A1=";A1,"Q=";Q
1710 GOSUB 2000:GOTO 1110
1750 INPUT "A1=";A1,"AN=";AN,"SN=";SN
1760 Q=(SN-A1)/(SN-AN)
1770 PRINT "Q=";Q:GOTO 1110
1800 INPUT "N=";N,"AN=";AN,"A1=";A1
1810 Q=(AN/A1)ROT(N-1)
1820 GOTO 1770
1850 INPUT "A1=";A1,"Q=";Q,"N=";N
1860 SN=A1*((Q^N-1)/(Q-1))
1870 PRINT "SN=";SN:GOTO 1110
1900 INPUT "AN=";AN,"Q=";Q,"N=";N
1910 SN=(AN/Q^(N-1))*((Q^N-1)/(Q-1))
1920 GOTO 1870
1950 INPUT "AN=";AN,"A1=";A1,"Q=";Q
1960 SN=(Q*AN-A1)/(Q-1):GOTO 1870
1970 INPUT "AN=";AN,"A1=";A1,"N=";N
1980 SN=((AN/A1)ROT(N-1)*AN-A1)/(((AN/A1)ROT(N-1))-1)
1990 GOTO 1870
2000 N=(LOG AN-LOG A1+LOG Q)/(LOG Q)
2010 PRINT "N=";N:RETURN
2020 INPUT "A1=";A1,"SN=";SN,"Q=";Q
2030 AN=A1*Q^((LOG((SN/A1)*(Q-1)+1)/(LOG Q))-1)
2040 PRINT "AN=";AN:GOTO 1110

```

Schiefe Würfe

Beschreibung:

Dieses Programm dient zur Berechnung von schiefen Würfeln. Es ist hinreichend bekannt, daß es heutzutage immer noch viele Physiklehrer gibt, deren Lieblingsbeschäftigung darin besteht die Schüler mit der Berechnung von eben diesen schiefen Würfeln zu beschäftigen.

DAS PROGRAMM HAT 6 VERSCHIEDENE TEILE:

- FLUGHÖHENBERECHNUNG	DEF C
- FLUGWEITEN	DEF H
- ABWURFWINKEL	DEF J
- ANFANGSGESCHWINDIGKEITSBERECHNUNG	DEF SPC
- SCHWERKRAFT	DEF L
- FLUGDAUER	DEF Z

VORSICHT IST JEDOCH GEBOTEN, WENN ABWURFHÖHE UND AUFPRALLHÖHE NICHT IN EINER EBENE LIEGEN !

Listing:

```

5000 "C":PAUSE "PHYSIK SY-BERECH"
5010 PAUSE "WERTE EINGEBEN"
5020 INPUT "SX=";A
5030 INPUT "WINKEL=";B
5040 INPUT "G=";C
5050 INPUT "V0=";D
5060 E=A*TAN(B):F=C*A*A:G=2*D*D*(COS(B)^2):H=E-(F/G)
5070 PRINT "SY=";H
5080 END
5090 "H":PAUSE "PHYSIK SX-BER."
5100 PAUSE "WERTE EINGEBEN"
5110 INPUT "SY=";E
5120 INPUT "WINKEL=";B
5130 INPUT "G=";C
5140 INPUT "V0=";D
5150 F=(4*D*D*D*D*(TAN(B)^2)*(COS(B)^4))/(4*C*C)
5160 G=(E*2*D*D*(COS(B)^2))/C
5170 H=(2*D*D*TAN(B)*(COS(B)^2))/(2*C)
5180 I=(SQR(F-G))+H:J=-SQR(F-G)+H
5190 BEEP 2:PAUSE "2 WERTE ";PAUSE "SX1=":PRINT I:
    PAUSE "SX2=":PRINT J:END
5200 "J":PAUSE "WINKEL BER."
5210 BEEP 1:PAUSE "NUR FUER HMAX!"
5220 PAUSE "WERTE EINGEBEN"
5230 INPUT "SY MAX=";E
5240 INPUT "V0=";D
5250 INPUT "G=";C
5260 F=ASN SQR((E*2*C)/(D*D))
5270 PRINT "WINKEL=";F
5280 END
5290 " ":PAUSE "PHY. VO BERECH."
5300 PAUSE "WERTE EINGEBEN"
5310 INPUT "SX=";A
5320 INPUT "WINKEL=";B
5330 INPUT "G=";C
5340 INPUT "SY=";E
5350 F=C*A*A:H=2*A*SIN(B)*COS(B)-(E*2*(COS(B)^2))
5360 I=SQR(F/H)
5370 PRINT "V0=";I
5380 END
5390 "L":PAUSE "PHYSIK G BERE."
5400 PAUSE "WERTE EINGEBEN"
5410 INPUT "SX=";A
5420 INPUT "WINKEL=";B
5430 INPUT "V0=";D

```

```
5440 INPUT "SY=";E
5450 F=((A*TAN(B)-E)*(2*D*D*(COS(B)^2)))
5460 G=F/(A*A)
5470 PRINT "G=";G
5480 END
5490 "Z":PAUSE "PHYSIK T BER."
5500 PAUSE "WERTE EINGEBEN"
5510 INPUT "V0=";D
5520 INPUT "WINKEL=";B
5530 INPUT "G=";C
5540 F=(2*D*SIN(B))/C
5550 PRINT "T=";F
5560 END
```

Kapitel 4.2

Elektronik und EDV

Dezimal-Binär-Umwandlung

Beschreibung:

Dieses Programm wandelt nach der Eingabe einer Dezimalzahl (zwischen 0 und 255) diese in eine Binärzahl um. Nach der Eingabe einer Binärzahl (8 Bit), wird diese in eine Dezimalzahl umgewandelt. Der Rechner erkennt automatisch, um welche Art von Zahl es sich handelt, da Binärzahlen mit I und O und Dezimalzahlen mit den normalen Zifferntasten eingegeben werden.

Funktionsweise:

Nach dem Programmstart werden Sie aufgefordert eine Zahl einzugeben. Bei dieser Zahl kann es sich entweder um eine 8-stellige Binär- oder um eine Dezimalzahl zwischen 0 und 255 handeln. Nach «ENTER» erscheint dann die Dezimal- bzw. Binärzahl.

Beispiel:	Rechner:	Eingabe:
	"D/B-ZAHL="	IOIOIOIO
	(BEEP) "170"	
	"D/B-ZAHL="	170
	(BEEP) "IOIOIOIO"	

Listing:

```

10 DIM B$(2)*16
20 INPUT "D/B - ZAHL = "; B$(2); B$(1)="": IF LEFT$(B$(2), 1) =
  "I" OR LEFT$(B$(2), 1) = "O" THEN 100
25 A=VAL B$(2): RESTORE : IF A>255 THEN 20
30 READ D: IF D=0 THEN 80
40 C=A/D
50 IF C>=1 THEN LET B$(1)=B$(1)+"I ": GOTO 70
60 IF C<1 THEN LET B$(1)=B$(1)+"O ": GOTO 30
70 A=(C-INT(C))*D: GOTO 30
80 BEEP 1: WAIT 250: PRINT B$(1): GOTO 20
90 DATA 128, 64, 32, 16, 8, 4, 2, 1, 0
100 A=0: RESTORE : IF LEN B$(2)<>8 THEN 20
110 FOR N=1 TO 8: READ D
120 IF MID$(B$(2), N, 1) = "I" THEN LET A=A+D
125 NEXT N
130 B$(1)="          "+STR$ A: GOTO 80

```

Dezimal-Hexadezimal-Binär-Umwandlung

Beschreibung:

Hierbei handelt es sich um ein Umrechnungsprogramm, das es erlaubt in kurzer Zeit zwischen den drei gebräuchlichsten Zahlensystemen umzuwandeln.

Funktionsweise:

Nach starten des Programmes mit **RUN** fragt der Rechner in welchem Zahlensystem die Eingabe erfolgen soll. Hat man sich für ein Zahlensystem entschieden und selbiges ausgewählt, erfolgt die Eingabe des Zahlenwertes. Nun erfolgt automatisch die Ausgabe aller Werte in den verschiedenen Zahlensystemen.

Wichtig hierbei ist, daß Listing 1 für die SHARP-PC's 1350/60 erstellt wurde und Listing 2 für den SHARP-PC 1403 erstellt wurde.

Listing 1:

```

10 "A":CLEAR:CLS:WAIT 0
15 PRINT ">>UMRECHNUNGSPROGRAMM<<"
16 CURSOR 0,1:PRINT "-----"
20 CURSOR 5,2:PRINT "Eingabe in ???"
25 CURSOR 3,3:PRINT "DEZ  HEX  BIN"
30 A$=INKEY$
35 IF A$="D" THEN 1000
40 IF A$="H" THEN 2000
45 IF A$="B" THEN 3000
50 GOTO 30
1000 CLEAR:CLS:DIM HE$(4)*1:DIM H(4)
1005 PRINT "EINGABE DEZIMAL"
1010 CURSOR 0,1:INPUT "DEZ ";DEZ
1015 IF DEZ<>INT(DEZ)THEN 1000
1020 IF DEZ>65535 THEN 1000
1025 IF DEZ<0 THEN 1000
1030 A=DEZ/256
1035 M=INT(A):L=DEZ-M*256
1040 A=M/16
1045 H(1)=INT(A):H(2)=M-H(1)*16
1050 A=L/16
1055 H(3)=INT(A):H(4)=L-H(3)*16
1065 FOR A=1 TO 4
1070 IF H(A)=15 THEN LET HE$(A)="F"
1075 IF H(A)=14 THEN LET HE$(A)="E"
1080 IF H(A)=13 THEN LET HE$(A)="D"
1085 IF H(A)=12 THEN LET HE$(A)="C"
1090 IF H(A)=11 THEN LET HE$(A)="B"
1095 IF H(A)=10 THEN LET HE$(A)="A"
1100 IF H(A)<10 THEN LET HE$(A)=STR$(H(A))
1105 NEXT A
1110 CURSOR 0,2:PRINT "HEX ";HE$(1);HE$(2);HE$(3);HE$(4)
1115 IF DEZ>255 THEN 1200
1120 LET BIN=DEZ
1125 IF BIN>127 THEN LET BIN=BIN-128:LET B7=1
1130 IF BIN>63 THEN LET BIN=BIN-64:LET B6=1
1135 IF BIN>31 THEN LET BIN=BIN-32:LET B5=1
1140 IF BIN>15 THEN LET BIN=BIN-16:LET B4=1
1145 IF BIN>7 THEN LET BIN=BIN-8:LET B3=1
1150 IF BIN>3 THEN LET BIN=BIN-4:LET B2=1
1155 IF BIN>1 THEN LET BIN=BIN-2:LET B1=1
1160 LET B0=BIN:WAIT
1165 CURSOR 0,3:PRINT "BIN ";B7;B6;B5;B4;" ";B3;B2;B1;B0
1170 GOTO 10
1200 WAIT
1205 CURSOR 0,3:PRINT "BIN zu gross":GOTO 10

```

```

2000 CLEAR:CLS:A=2000:DIM HE$(5)
2010 PRINT "EINGABE HEXADEZIMAL"
2015 INPUT "HEX ";HE$
2017 IF LEN(HE$)>4 THEN A
2020 LET H4=-48+ASC(MID$(HE$,1,1))
2025 LET H3=-48+ASC(MID$(HE$,2,1))
2030 LET H2=-48+ASC(MID$(HE$,3,1))
2035 LET H1=-48+ASC(MID$(HE$,4,1))
2040 IF H4>9 THEN LET H4=H4-7
2045 IF H3>9 THEN LET H3=H3-7
2050 IF H2>9 THEN LET H2=H2-7
2055 IF H1>9 THEN LET H1=H1-7
2060 IF H4=-48 THEN A
2065 IF H3=-48 THEN LET H1=H4:H2=00:H4=0:H3=0
2070 IF H2=-48 THEN LET H2=H4:H1=H3:H4=0:H3=0
2075 IF H1=-48 THEN LET H1=H2:H2=H3:H3=H4:H4=0
2080 IF H4>15 THEN A
2085 IF H3>15 THEN A
2090 IF H2>15 THEN A
2095 IF H1>15 THEN A
2100 IF H4<0 THEN A
2105 IF H3<0 THEN A
2110 IF H2<0 THEN A
2115 IF H1<0 THEN A
2120 DEZ=H4*4096+H3*256+H2*16+H1
2125 CURSOR 0,2:PRINT "DEZ ";DEZ
2130 IF DEZ>255 THEN 2200
2135 GOTO 1120
2200 WAIT
2205 CURSOR 0,3:PRINT "BIN zu gross":GOTO 10
3000 CLEAR:CLS:A=3000:DIM BINS(9):DIM B(8):DIM HE$(2)*1
3005 PRINT "EINGABE BINAER"
3010 CURSOR 0,1:INPUT "BIN ";BINS$
3015 IF LEN(BINS$)>8 THEN A
3020 LET Y=LEN(BINS$)
3025 FOR X=9-Y TO 8
3026 W=W+1
3030 LET B(X)=-48+ASC(MID$(BINS$,W,1))
3035 IF B(X)<0 THEN LET Z=3
3040 IF B(X)>1 THEN LET Z=3
3045 NEXT X
3050 IF Z=3 THEN A
3055 IF B(1)=1 THEN LET DEZ=128
3060 IF B(2)=1 THEN LET DEZ=DEZ+64
3065 IF B(3)=1 THEN LET DEZ=DEZ+32
3070 IF B(4)=1 THEN LET DEZ=DEZ+16
3075 IF B(5)=1 THEN LET DEZ=DEZ+8
3080 IF B(6)=1 THEN LET DEZ=DEZ+4
3085 IF B(7)=1 THEN LET DEZ=DEZ+2
3090 IF B(8)=1 THEN LET DEZ=DEZ+1
3095 CURSOR 0,2:PRINT "DEZ ";DEZ
3100 X=INT(DEZ/16):Y=DEZ-(X*16)
3105 IF X=15 THEN LET HE$(1)="F"
3110 IF X=14 THEN LET HE$(1)="E"
3115 IF X=13 THEN LET HE$(1)="D"
3120 IF X=12 THEN LET HE$(1)="C"
3125 IF X=11 THEN LET HE$(1)="B"
3130 IF X=10 THEN LET HE$(1)="A"
3135 IF X<10 THEN LET HE$(1)=STR$(X)

```

```

3150 IF Y=15 THEN LET HE$(2)="F"
3155 IF Y=14 THEN LET HE$(2)="E"
3160 IF Y=13 THEN LET HE$(2)="D"
3165 IF Y=12 THEN LET HE$(2)="C"
3170 IF Y=11 THEN LET HE$(2)="B"
3175 IF Y=10 THEN LET HE$(2)="A"
3180 IF Y<10 THEN LET HE$(2)=STR$(Y)
3185 WAIT
3190 CURSOR 0,3:PRINT "HEX ";HE$(1);HE$(2)
3200 GOTO 10

```

Listing 2:

```

10 "A"
15 PRINT ">>UMRECHNUNGSPROGRAMM<<"
20 PRINT "EINGABE IN ????"
25 PRINT "DEZ  HEX  BIN"
30 A$=INKEY$
35 IF A$="D" THEN 1000
40 IF A$="H" THEN 2000
45 IF A$="B" THEN 3000
50 GOTO 30
1000 CLEAR:DIM HE$(4)*1:DIM H(4)
1005 PRINT "EINGABE DEZIMAL"
1010 INPUT "DEZ ";DEZ
1015 IF DEZ<>INT(DEZ)THEN 1000
1020 IF DEZ>65535 THEN 1000
1025 IF DEZ<0 THEN 1000
1030 A=DEZ/256
1035 M=INT(A):L=DEZ-M*256
1040 A=M/16
1045 H(1)=INT(A):H(2)=M-H(1)*16
1050 A=L/16
1055 H(3)=INT(A):H(4)=L-H(3)*16
1065 FOR A=1 TO 4
1070 IF H(A)=15 THEN LET HE$(A)="F"
1075 IF H(A)=14 THEN LET HE$(A)="E"
1080 IF H(A)=13 THEN LET HE$(A)="D"
1085 IF H(A)=12 THEN LET HE$(A)="C"
1090 IF H(A)=11 THEN LET HE$(A)="B"
1095 IF H(A)=10 THEN LET HE$(A)="A"
1100 IF H(A)<10 THEN LET HE$(A)=STR$(H(A))
1105 NEXT A
1110 PRINT "HEX ";HE$(1);HE$(2);HE$(3);HE$(4)
1115 IF DEZ>255 THEN 1200
1120 LET BIN=DEZ
1125 IF BIN>127 THEN LET BIN=BIN-128:LET B7=1
1130 IF BIN>63 THEN LET BIN=BIN-64:LET B6=1
1135 IF BIN>31 THEN LET BIN=BIN-32:LET B5=1
1140 IF BIN>15 THEN LET BIN=BIN-16:LET B4=1
1145 IF BIN>7 THEN LET BIN=BIN-8:LET B3=1
1150 IF BIN>3 THEN LET BIN=BIN-4:LET B2=1
1155 IF BIN>1 THEN LET BIN=BIN-2:LET B1=1
1160 LET B0=BIN:WAIT
1165 PRINT "BIN ";B7;B6;B5;B4;" ";B3;B2;B1;B0
1170 GOTO 10
1200 WAIT
1205 PRINT "BIN ZU GROSS":GOTO 10

```

```
2000 CLEAR:A=2000:DIM HES(5)
2010 PRINT "EINGABE HEXADEZIMAL"
2015 INPUT "HEX ";HES
2017 IF LEN(HES)>4 THEN A
2020 LET H4=-48+ASC(MID$(HES,1,1))
2025 LET H3=-48+ASC(MID$(HES,2,1))
2030 LET H2=-48+ASC(MID$(HES,3,1))
2035 LET H1=-48+ASC(MID$(HES,4,1))
2040 IF H4>9 THEN LET H4=H4-7
2045 IF H3>9 THEN LET H3=H3-7
2050 IF H2>9 THEN LET H2=H2-7
2055 IF H1>9 THEN LET H1=H1-7
2060 IF H4=-48 THEN A
2065 IF H3=-48 THEN LET H1=H4:H2=00:H4=0:H3=0
2070 IF H2=-48 THEN LET H2=H4:H1=H3:H4=0:H3=0
2075 IF H1=-48 THEN LET H1=H2:H2=H3:H3=H4:H4=0
2080 IF H4>15 THEN A
2085 IF H3>15 THEN A
2090 IF H2>15 THEN A
2095 IF H1>15 THEN A
2100 IF H4<0 THEN A
2105 IF H3<0 THEN A
2110 IF H2<0 THEN A
2115 IF H1<0 THEN A
2120 DEZ=H4*4096+H3*256+H2*16+H1
2125 PRINT "DEZ ";DEZ
2130 IF DEZ>255 THEN GOTO 2200
2135 GOTO 1120
2200 WAIT
2205 PRINT "BIN ZU GROSS":GOTO 10
3000 CLEAR:A=3000:DIM BIN$(9):DIM B(8):DIM HES(2)*1
3005 PRINT "EINGABE BINAER"
3010 INPUT "BIN ";BIN$
3015 IF LEN(BIN$)>8 THEN A
3020 LET Y=LEN(BIN$)
3025 FOR X=9-Y TO 8
3026 W=W+1
3030 LET B(X)=-48+ASC(MID$(BIN$,W,1))
3035 IF B(X)<0 THEN LET Z=3
3040 IF B(X)>1 THEN LET Z=3
3045 NEXT X
3050 IF Z=3 THEN A
3055 IF B(1)=1 THEN LET DEZ=128
3060 IF B(2)=1 THEN LET DEZ=DEZ+64
3065 IF B(3)=1 THEN LET DEZ=DEZ+32
3070 IF B(4)=1 THEN LET DEZ=DEZ+16
3075 IF B(5)=1 THEN LET DEZ=DEZ+8
3080 IF B(6)=1 THEN LET DEZ=DEZ+4
3085 IF B(7)=1 THEN LET DEZ=DEZ+2
3090 IF B(8)=1 THEN LET DEZ=DEZ+1
3095 PRINT "DEZ ";DEZ
3100 X=INT(DEZ/16):Y=DEZ-(X*16)
3105 IF X=15 THEN LET HES(1)="F"
3110 IF X=14 THEN LET HES(1)="E"
3115 IF X=13 THEN LET HES(1)="D"
3120 IF X=12 THEN LET HES(1)="C"
3125 IF X=11 THEN LET HES(1)="B"
3130 IF X=10 THEN LET HES(1)="A"
3135 IF X<10 THEN LET HES(1)=STR$(X)
```



```

3150 IF Y=15 THEN LET HE$(2)="F"
3155 IF Y=14 THEN LET HE$(2)="E"
3160 IF Y=13 THEN LET HE$(2)="D"
3165 IF Y=12 THEN LET HE$(2)="C"
3170 IF Y=11 THEN LET HE$(2)="B"
3175 IF Y=10 THEN LET HE$(2)="A"
3180 IF Y<10 THEN LET HE$(2)=STR$(Y)
3185 WAIT
3190 PRINT "HEX ";HE$(1);HE$(2)
3200 GOTO 10

```

Hamming-Code

Beschreibung:

Mit Hilfe des Hamming-Code ist es möglich, digitale Informationen so zu codieren, daß bei einem Übertragungsfehler das fehlerhafte Bit lokalisiert und korrigiert werden kann. Die Fehlererkennung wird durch Verwendung mehrerer Prüfbits durchgeführt. Formel zur Berechnung der benötigten Prüfbits bei m Informationsbits:

k = Anzahl der Prüfbits
 m = Anzahl der Infobits

$$2^k \geq m+k+1$$

Daraus folgt, daß bei der Codierung von 11 Informationsbits 4 Prüfbits benötigt werden, die man folgendermaßen ermittelt: Die Quersumme der Inhalte der Informationsbits die jeweils ein Prüfbit beeinflussen muß gerade sein, damit das Prüfbit 0 wird. D.h., das Prüfbit ergänzt auf gerade Parität. Das Prüfbit wird von den jeweils dahinterstehenden Informationsbits beeinflusst:

k1 :	m1	m2	m4	m5	m7	m9	m11
k2 :	m1	m3	m4	m6	m7	m10	m11
k3 :	m2	m3	m4	m8	m9	m10	m11
k4 :	m5	m6	m7	m8	m9	m10	m11

Das Bildungsgesetz nachdem die Abhängigkeit festgelegt wird, sowie das lokalisieren eines Fehlers, ist in der entsprechenden Fachliteratur nachzulesen, da dies hier zu weit führen würde. Die Ausgabe der codierten Information erfolgt in folgender Reihenfolge:

m11 m10 m9 m8 m7 m6 m5 k4 m4 m3 m2 k3 m1 k2 k1

Um nun z.B. die Information

m8	m7	m6	m5	m4	m3	m2	m1
0	0	0	0	1	0	1	1

zu verschlüsseln geht man folgendermaßen vor:

Bildung von k1: Es werden m1 m2 m4 m5 m7 m9 m11 addiert
 Ergebnis: 3 ==> k1 muß 1 werden um auf gerade Parität
 (Anzahl von Einsen) zu ergänzen.

Mit k2...k4 verfährt man ebenso und erhält dann die codierte Information von:

```

m11 m10 m9 m8 m7 m6 m5 k4 m4 m3 m2 k3 m1 k2 k1
  0   0   0   0   0   0   0   0   1   0   1   0   1   0   1

```

Damit diese aufwendige bzw. zeitraubende Codierung nicht ständig manuell durchgeführt werden muß, wurde das nachstehende Programm entwickelt. Es werden Informationslängen bis zu 11 Bit codiert, da bei der Ausgabe noch 4 Prüfbits hinzukommen und der 1401/1402 nur eine Anzeige von max. 16 Stellen hat. Wird dem Rechner eine Information mit weniger als 11 Bit gegeben, so werden die restlichen von vorne her mit 0 aufgefüllt, was ja die Information nicht verändert. Nach dem Start meldet sich der Rechner mit "INFORMATION:". Nun tippen Sie die Informationsbits ein. Nach etwas Rechenzeit ist ein BEEP zu hören und das Ergebnis erscheint im Display. Die führenden Nullen können je nach Bedarf weggelassen werden.

Beispiel:	Eingabe:	Ausgabe:
	10101010101	101010100101101

Listing:

```

10 DIM K(4),MM$(11),KK$(4),M$(11)
20 RESTORE : INPUT "INFORMATION: ";MM$
30 IF LEN(MM$)>11 THEN 20
40 IF LEN(MM$)<11 THEN LET MM$="0"+MM$: GOTO 40
50 FOR N=1 TO 11:M$(N)=MID$(MM$,12-N,1): NEXT N
60 K(1)=0:K(2)=0:K(3)=0:K(4)=0:KK$(1)="1":KK$(2)="1"
   KK$(3)="1":KK$(4)="1"
70 FOR N=1 TO 4
80 FOR O=1 TO 7
90 READ B$
100 K(N)=K(N)+VAL(M$(VAL(B$)))
110 NEXT O
120 IF INT(K(N)/2)=K(N)/2 THEN LET KK$(N)="0"
130 NEXT N
140 MM$=""
150 MM$=M$(11)+M$(10)+M$(9)+M$(8)+M$(7)+M$(6)
   +M$(5)+KK$(4)+M$(4)+M$(3)
160 MM$=MM$+M$(2)+KK$(3)+M$(1)+KK$(2)+KK$(1)
170 BEEP 1: PRINT MM$
180 GOTO 20
200 DATA "1","2","4","5","7","9","11"
210 DATA "1","3","4","6","7","10","11"
220 DATA "2","3","4","8","9","10","11"
230 DATA "5","6","7","8","9","10","11"

```

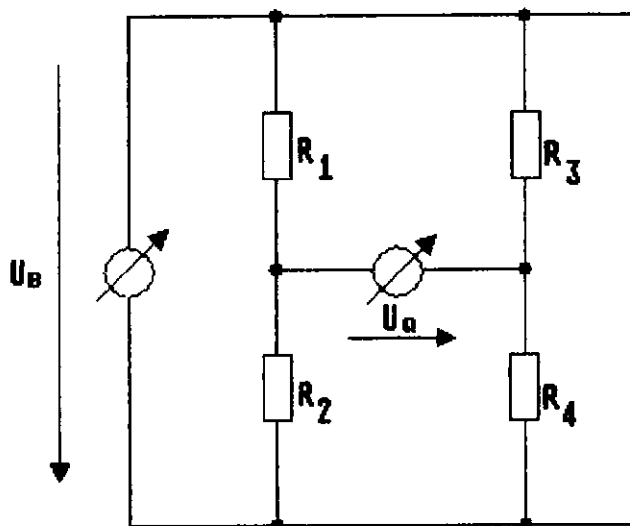
Wheatstone Brücke

Beschreibung:

Mit dem nachstehenden Programm kann eine Brückenschaltung nach untenstehendem Schaltplan berechnet werden. Zuerst werden Sie vom Programm nach den einzelnen Widerständen gefragt. Gibt man bei einer der Widerstandsabfragen nur «ENTER» ein, wird dieser Widerstand so berechnet, daß die Brücke abgeglichen ist - werden jedoch alle 4 Widerstände eingegeben, so wird mit diesen Widerständen gerechnet (auch in unabgeglichenem Zustand). Im Falle der Berechnung eines Widerstandes, wird dessen Wert nach diesen Abfragen angezeigt. Danach erfolgt die Eingabe der Betriebsspannung UB. Ist diese eingetippt wird die Querspannung UQ angezeigt (im abgeglichenen Zustand = 0). Danach zeigt Ihnen der Rechner die Brückenstromströme, sowie die notwendige bzw. verbrauchte Leistung der Widerstände an. Der Spannungsmesser V wird als ideal betrachtet ($R_i \Rightarrow \infty$). Bei diesem Programm wird ein kleiner Programmiertrick benutzt um festzustellen, welcher Widerstand berechnet werden soll. Kommt der Rechner in Zeile 30 an, verlangt er die Eingabe des R1 - wird jetzt nur ein «ENTER» eingegeben, bricht der Rechner diese Zeile sofort ab und springt in die nächste Zeile (Zeile 40) - hier wird der Variablen A\$ der Name des unbekanntes Widerstandes mitgeteilt - gibt man jedoch in Zeile 30 eine Zahl mit darauffolgendem «ENTER» ein, wird die Zeile ordnungsgemäß ausgeführt und es kommt zu einem Sprung nach Zeile 50, d.h. es erfolgt keine Zuweisung in Zeile 40.

Beispiel:

Anzeige:	Eingabe:	
WHEATSTONEBR.		«ENTER»
R1 IN OHM?	1000	«ENTER»
R2 IN OHM?	2000	«ENTER»
R3 IN OHM?	3000	«ENTER»
R4 IN OHM?	4000	«ENTER»
UB IN VOLT?	15	«ENTER»
UQ IN VOLT:		«ENTER»
-1.428571429		«ENTER»
I1/2 IN A:		«ENTER»
0.005		«ENTER»
I3/4 IN A:		«ENTER»
2.142857143E-03		«ENTER»
P1 IN WATT:		«ENTER»
0.025		«ENTER»
P2 IN WATT:		«ENTER»
0.05		«ENTER»
P3 IN WATT:		«ENTER»
1.377551021E-02		«ENTER»
P4 IN WATT:		«ENTER»
1.836734694E-02		«ENTER» ==> NEUSTART

SCHALTPLAN:Listing:

```

10 PRINT "WHEATSTONEBR."
20 CLEAR
30 INPUT "R1 IN OHM? ";R1: GOTO 50
40 A$=A$+"R1"
50 INPUT "R2 IN OHM? ";R2: GOTO 70
60 A$=A$+"R2"
70 INPUT "R3 IN OHM? ";R3: GOTO 90
80 A$=A$+"R3"
90 INPUT "R4 IN OHM? ";R4: GOTO 105
100 A$=A$+"R4"
105 IF A$="" THEN 130
110 IF LEN A$<>2 THEN 20
120 GOSUB A$: PRINT A$;" IN OHM:": PRINT B
130 INPUT "UB IN VOLT? ";UB
140 UQ=UB*(R1/(R1+R2)-R3/(R3+R4))
150 PRINT "UQ IN VOLT :": PRINT UQ
160 PRINT "I1/2 IN A:": I1=UB/(R1+R2): PRINT I1
170 PRINT "I3/4 IN A:": I3=UB/(R3+R4): PRINT I3
180 PRINT "P1 IN WATT:": PRINT I1^2*R1
190 PRINT "P2 IN WATT:": PRINT I1^2*R2
200 PRINT "P3 IN WATT:": PRINT I3^2*R3
210 PRINT "P4 IN WATT:": PRINT I3^2*R4
220 GOTO 20
250 "R1"R1=R2*R3/R4:B=R1: RETURN
260 "R2"R2=R1*R4/R3:B=R2: RETURN
270 "R3"R3=R1*R4/R2:B=R3: RETURN
280 "R4"R4=R2*R3/R1:B=R4: RETURN

```

Einfaches Schaltinterface

Beschreibung:

Es ist bestimmt schon mancher auf die Idee gekommen mit seinem Rechner etwas ein- bzw. auszuschalten. Für diejenigen die einen Drucker CE-126 besitzen, ist es jetzt kein Problem mehr, solche Anwendungen zu realisieren. Mit CALL 32831 (&803F) kann das Relais des Druckers ein- und mit CALL 32834 (&8042) wieder ausgeschaltet werden.

So kann z.B. mit Hilfe des Programms TIMER die Abschaltung eines Ladegeräts für NiCa-Zellen erfolgen. Hierzu ist in das Programm noch eine Abbruchbedingung einzufügen, welche nach Beendigung der gewünschten Zeit CALL 32834 aufruft.

Nachstehendes Programm wurde speziell dazu entwickelt, NiCa-Zellen zu laden und bei genügender Ladezeit das Ladegerät von den Zellen zu trennen, sowie den Rechner abzuschalten. Das Programm benötigt die Akku-Kapazität und den Ladestrom in mA. Aus diesen beiden Werten wird dann die benötigte Ladezeit berechnet. Während des Ladevorgangs wird die noch benötigte Ladezeit in Minuten im Display des Rechners angezeigt um eine Kontrolle zu gewährleisten.

Zuerst werden Sie vom Programm nach der Akku-Kapazität gefragt. Nach der Eingabe in mAh (z.B. 500) erfolgt die Frage nach dem Ladestrom in mA (z.B. 100). Nun wird vom Rechner die normalerweise benötigte Ladezeit berechnet und angezeigt. Danach blinkt "RICHTIG (J/N/Z)". Wählt man "J" wird das Gerät eingeschaltet und das Laden beginnt - wird "N" gewählt, so kann man manuell eine gewünschte Zeit in Minuten eingeben (z.B. wenn die Akkus nicht ganz leer sind) - bei "Z" wird nochmals die berechnete Zeit angezeigt.

Ist der Ladevorgang beendet, d.h. die Zeit auf 0, so wird das Gerät abgeschaltet, der Rechner piept und schaltet sich ebenfalls ab.

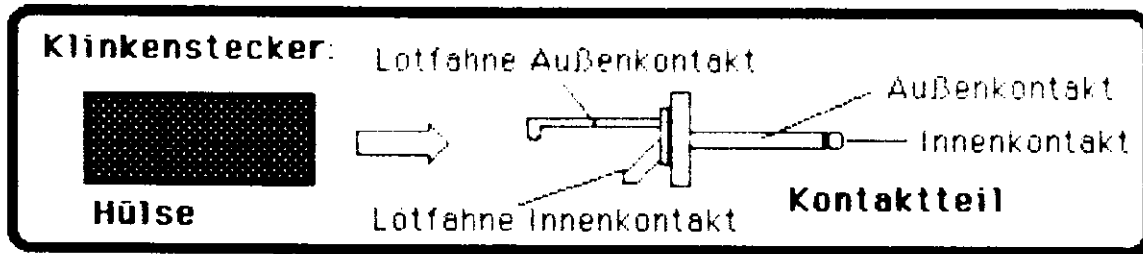
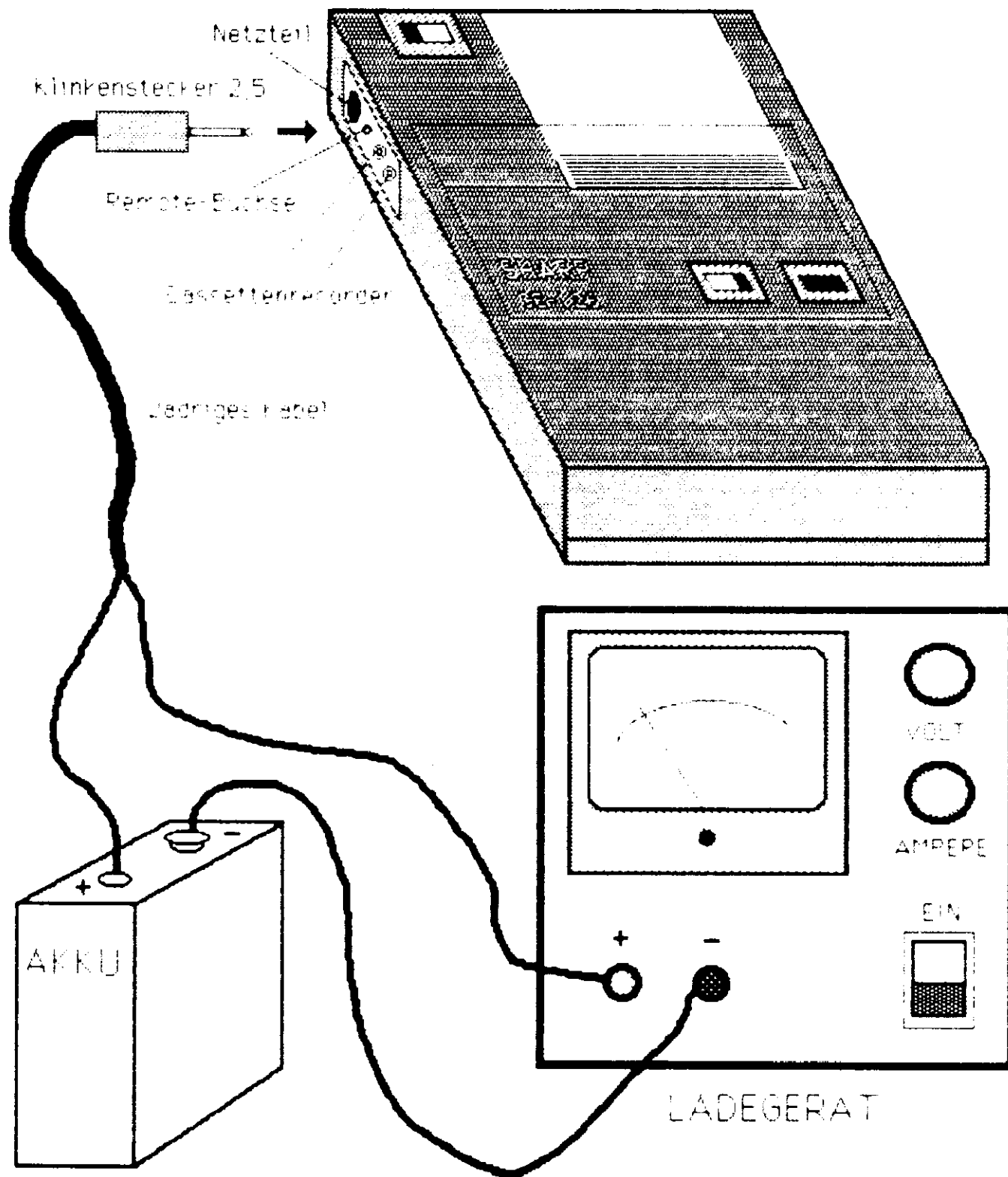
WICHTIGER HINWEIS: Das Relais des CE-126 kann nur Ströme bis maximal 500 mA bei 12V verkraften! Wenn größere Ströme geschaltet werden sollen, so ist noch ein zusätzliches Relais dazwischenzuschalten! Mit diesem einfachen Interface dürfen keinesfalls Netzspannungen geschaltet werden!

Listing:

```

10 CLEAR
20 INPUT "KAPAZITAET: ";AK
30 INPUT "LADESTROM: ";LA
40 IF AK=0 OR LA=0 THEN 10
50 A= INT (AK/LA*84)
60 WAIT 50: PRINT "ZEIT: "; STR$(A);" MIN"
70 WAIT 10: PRINT "RICHTIG (J/N/Z)"
80 B$= INKEY$: IF B$<>"J" AND B$<>"N" AND B$<>"Z" THEN 70
90 IF B$="N" THEN INPUT "LADEZEIT (MIN):";A:A= INT (A)
100 IF B$="Z" THEN 60
110 IF B$<>"J" THEN 70
120 A= ABS (A)
130 CALL &803F
140 WAIT 3527
150 PRINT A:A=A-1: IF A=0 THEN CALL &8042: BEEP 2:
    CALL 1456
160 GOTO 150

```

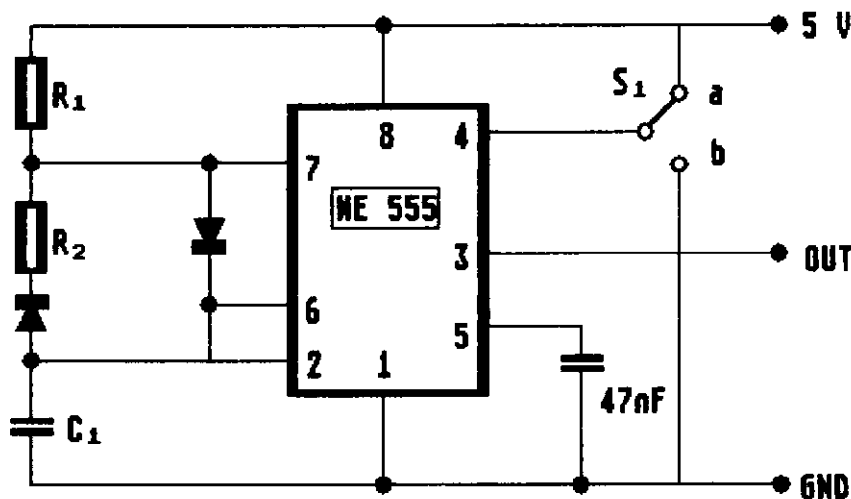


OSZILLATOR NE 555

Beschreibung:

Der Schaltkreis NE 555 (oder LM 555) ist die Standardschaltung der modernen Elektronik zum Erzeugen von Rechteckimpulsen. Der Schaltkreis NE 555 ist TTL-kompatibel, kann einen hohen Ausgangsstrom liefern, das Tastverhältnis kann frei gewählt werden und der Oszillator kann mit einem Steuersignal ein- und wieder ausgeschaltet werden. Typische Anwendungen sind TTL-Oszillatoren oder z.B. Blinkschaltungen. Nachfolgend der Schaltplan der Standard-Applikation.

SCHALTPLAN:



Diden = 1N4148

Hinweise zur Schaltung:

Die Versorgungsspannung darf zwischen 4,5 V und 15 V liegen und der maximale Ausgangsstrom ist auf 200 mA zu begrenzen. Für maximale Frequenzstabilität sollte man nur Metallfilmwiderstände und Folienkondensatoren (z.B. WIMA MKC) verwenden. Bei Kondensatoren über 1µF sind Tantalkondensatoren zu verwenden.

Die HIGH-Zeit des Ausgangsimpulses hängt von R1, die LOW-Zeit von R2 ab.

Programmhinweis:

Bevor das Programm die Eingabe von C1 erfragt, wird vom Programm ein Tip gegeben, in welcher Größenordnung sich der Wert von C1 bewegen sollte. Erweist sich der eingegebene Wert als unbrauchbar, so wird die Abfrage wiederholt.

Das Programm erfordert folgende Eingaben:

Versorgungsspannung	UE
Frequenz	F
HIGH-Zeit in	%
Kondensator	C1 (nach Vorschlag)

Es wird vom Programm ausgegeben:

Periodendauer in ms
 HIGH-Zeit in ms
 LOW-Zeit in ms
 Wert von C1
 Wert R1 und R2

Listing:

```

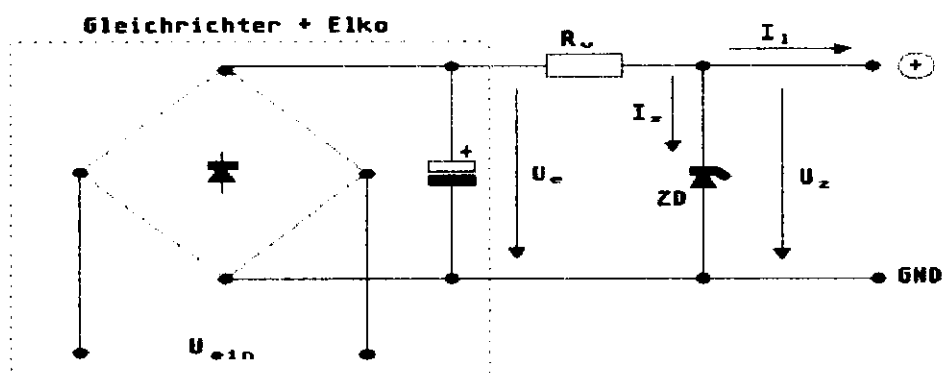
5 CLEAR
10 BEEP 1: PRINT "*555 OSZILATOR *"
20 INPUT "UB IN V:";UE: IF UE>16 OR UE<4.5 THEN 20
30 INPUT "FRQ IN HZ:";F
40 INPUT "HI IN %:";P
50 IF F>=1E6 THEN LET IE=0.025
60 IF F<1E6 AND F>1E3 THEN LET IE=0.005
70 IF F<=1E3 THEN LET IE=0.001
80 T=1/F:HT=T*P/100:LT=T-HT
85 R2=2*UE/(3*IE)
90 C1=LT*1E9/(R2* LOG ((2*UE/3-0.65)/(UE/3-0.65)))
100 PRINT "WAHL V. C1 IN NF": PRINT "RICHTWERT FOLGT":
    PRINT C1;"NF": INPUT C1
110 C1=C1/1E9:R1=HT/(C1* LOG ((2*UE/3-.65)/(UE/3-.65)))
120 R2=LT/(C1* LOG ((2*UE/3-.65)/(UE/3-.65)))
130 IF UE/R1>.05 THEN LET IE=IE/2: GOTO 85
140 BEEP 1: PRINT "ERGEBNISSE:": PRINT "T(MS)=";T*1000
150 PRINT "HI(MS)=";HT*1000
160 PRINT "LO(MS)=";LT*1000
170 PRINT "C1(NF)=";C1*1E9
180 PRINT "R1(OHM)=";R1: PRINT "R2(OHM)=";R2
190 INPUT "NOCHMAL ? J/N";XS
200 IF XS="J" THEN 10
210 END
  
```

Zenerdiode

Beschreibung:

Das vorliegende Programm berechnet den Vorwiderstand einer Standard-schaltung zur Spannungsstabilisierung mittels Zenerdiode. Solch eine Schaltung ist vor allem für kleine Leistungen geeignet, wie z.B. Referenzspannungsquelle oder zur Entkopplung von Versorgungsspannungen.

SCHALTPLAN:



Noch ein Tip:

Wenn an die Schaltung hohe Anforderungen bezüglich Rauschen und Restbrummen gestellt werden, so kann dies durch parallelschalten eines Kondensators von ca. 33nF bis 470nF zur Zenerdiode verbessert werden.

Funktionsweise:

Nach dem Starten des Programmes werden folgende Angaben vom Programm erwartet:

Zenerspannung	UZ in Volt
Leistung der Zenerdiode	PZ in Watt
Nennstrom	IZ der Zenerdiode in mA
Differentieller Widerstand	RD der Zenerdiode in Ohm
Maximale Eingangsspannung	UE
Minimale Eingangsspannung	UE
Maximaler Laststrom	IL
Minimaler Laststrom	IL

Anmerkung:

Der Nennstrom IZ ist der Strom, bei dem die Zenerspannung im Datenblatt angegeben wird. Er beträgt im allgemeinen ca. 5mA (für Zenerdioden mit 0,4W). Der differentielle Widerstand RD liegt bei kleineren Zenerdioden im allgemeinen zwischen 10 und 20 Ohm. Mit diesem Wert kann gerechnet werden, wenn der genaue Wert nicht bekannt ist.

Folgende Ergebnisse werden ausgegeben:

Ausgangsspannung	UA
Gemittelter Vorwiderstand	RV
Leistung des Vorwiderstandes in	Watt
Delta	UA
Minimaler und Maximaler Vorwiderstand	RV

Anmerkung:

Das Programm rechnet mit einem Sicherheitsfaktor von 1,2. Dieser kann im Programm gegebenenfalls abgeändert werden.

Listing:

```

520 PAUSE "<<< Z-DIODE >>>"
530 INPUT "Z-SPG UZ IN V:";UZ
540 INPUT "Z-POW PZ IN W:";PZ
550 INPUT "NENN-IZ IN MA:";IZ
555 IZ=IZ/1000
560 INPUT "RD IN OHM:";RD
570 INPUT "MAX. UE IN V:";UG
580 INPUT "MIN. UE IN V:";UK
590 IF UK<=1.2*UZ THEN PRINT "UE ZU KLEIN !": GOTO 520
600 IF UK>UG THEN 570
610 INPUT "MAX. IL IN A:";IG
620 INPUT "MIN. IL IN A:";IK
630 IF IK>IG THEN 610
640 P=UZ*(IZ+IG-IK)*1.2
650 IF P>PZ THEN PRINT "ZD LSTG ZU KLEIN": GOTO 520
660 RK=(UG-UZ)/(PZ/UZ+IK)
670 RG=(UK-UZ)/(IZ+IG)
680 DU=(UG-UK)*RD/((0.5*RK)+(0.5+RG))+RD*(IG-IK)
690 PV=(UG-UZ)*(IG+IZ)*1.2
700 RV=(RK+RG)/2

```

```

710 BEEP 1: PAUSE "ERGEBNISSE:"
720 PRINT "UA(V)=";UZ
730 PRINT "RV(OHM)=";RV
735 PRINT "P/RV(W)=";PV
738 USING "&&&&&&&##.##^"
740 PRINT "DUA(V)=";DU
745 USING
750 PRINT "MIN.RV=";RK
760 PRINT "MAX.RV=";RG
770 INPUT "NOCHMAL ? J/N";XS
780 IF XS="J" THEN 520
790 END

```

Stern-Dreieck-Umwandlung

Beschreibung:

Bei der Berechnung von Widerstandsnetzwerken ist es oftmals notwendig, eine Stern- in eine Dreieck- oder eine Dreieck- in eine Sternschaltung umzuwandeln. Zuerst werden Sie vom Programm gefragt, welche Art von Umwandlung Sie wünschen. Hierbei bedeutet SD-Stern/Dreieck und DS-Dreieck/Stern. Nach der Eingabe werden Sie gebeten, den Wert des ersten Widerstands in Ohm einzugeben. Sind auch die verbleibenden 2 Werte eingetippt, werden die Widerstände der gewählten Schaltungsart ausgegeben. Das Programm zeigt die einfache und vorteilhafte Handhabung von Labels und deren Ansprung mit Hilfe einer Stringvariablen.

Beispiel:

Rechner:	Eingabe:	
SD ODER DS ?	SD	«ENTER»
R10 IN OHM:	10	«ENTER»
R20 IN OHM:	20	«ENTER»
R30 IN OHM:	30	«ENTER»
R12 IN OHM:		«ENTER»
36.66666667		«ENTER»
R13 IN OHM:		«ENTER»
55		«ENTER»
R23 IN OHM:		«ENTER»
110		

Listing:

```

10 CLEAR
20 INPUT "SD ODER DS ? ";A$
30 IF A$<>"SD" AND A$<>"DS" THEN 20
35 GOSUB A$
40 FOR N=1 TO 3
50 B$=A$+ STR$(N): GOSUB B$
60 PRINT C$;" IN OHM:": PRINT D
70 NEXT N: GOTO 10
80 "SD"
90 INPUT " R10 IN OHM: ";R1

```

```

100 INPUT " R20 IN OHM: ";R2
110 INPUT " R30 IN OHM: ";R3
120 RETURN
130 "DS"
140 INPUT " R12 IN OHM: ";R1
150 INPUT " R13 IN OHM: ";R2
160 INPUT " R23 IN OHM: ";R3
170 R=R1+R2+R3
180 RETURN
190 "SD1"CS="R12":D=R1*R2/R3+R1+R2: RETURN
200 "SD2"CS="R13":D=R1*R3/R2+R1+R3: RETURN
210 "SD3"CS="R23":D=R2*R3/R1+R2+R3: RETURN
220 "DS1"CS="R10":D=R1*R2/R: RETURN
230 "DS2"CS="R20":D=R1*R3/R: RETURN
240 "DS3"CS="R30":D=R2*R3/R: RETURN

```

Kondensatoren-Berechnung

Beschreibung:

Nach der Eingabe der Kondensatorwerte einer Reihenschaltung von Kondensatoren und der Eingabe der an dieser Schaltung anliegenden Gesamtspannung, berechnet das Programm die nachfolgenden Werte:

1. Gesamtkapazität
2. Einzelspannungen
3. Gesamtenergiegehalt der Schaltung
4. Einzelenergiegehalte der Kondensatoren

Das Programm ist so ausgelegt, daß max. eine Reihenschaltung von 20 Kondensatoren berechnet werden kann. Dies kann aber je nach Bedarf leicht geändert werden (DIM Anweisung in Zeile 10).

Funktionsweise:

Nach "KONDENSATOR-" "BERECHNUNGEN" werden Sie gefragt: "C IN MIC.FARAD:". Nach der Eingabe des ersten Kondensators in Microfarad, erscheint wieder "C IN MIC.FARAD:". Sind alle Kondensatoren eingegeben, geben Sie «0» «ENTER» ein und es wird Ihnen die Gesamtkapazität angezeigt. Nach «ENTER» verlangt der Rechner die anliegende Gesamtspannung. Ist die Eingabe abgeschlossen, berechnet er die Einzelspannungen, wobei U₁ die Spannung ist, die an dem Kondensator abfällt, den Sie als erstes eingetippt haben (usw.). Nach der Anzeige kann jeweils mit «ENTER» fortgefahren werden. Sind alle Spannungen angezeigt, wird der Gesamtenergiegehalt der Schaltung berechnet. Nach «ENTER» werden die Einzelenergiegehalte in der selben Reihenfolge wie die Spannungen im Display dargestellt.

Beispiel:

Rechner:	Eingabe:
"KONDENSATOR-"	
"BERECHNUNGEN"	
"C IN MIC.FARAD:"	5 «ENTER»
"C IN MIC.FARAD:"	90 «ENTER»
"C IN MIC.FARAD:"	15 «ENTER»
"C IN MIC.FARAD:"	0 «ENTER»
"C-GES. ="	
"3.6000000E-3"	«ENTER»
"U-GES. IN V :"	100 «ENTER»
"U1 ="	
"7.2000000E 01"	«ENTER»
"U2 ="	
"4.0000000E 00"	«ENTER»
"U3 ="	
"2.4000000E 01"	«ENTER»
"W-GES. ="	
"1.8000000E 01"	«ENTER»
"W1 ="	
"1.2960000E 01"	«ENTER»
"W2 ="	
"7.2000000E-01"	«ENTER»
"W3 ="	
"4.3200000E 00"	«ENTER»
">"	

Listing:

```

10 DIM C(22):N=1:CG=0: PAUSE " KONDENSATOR  ":
   PAUSE " BERECHNUNGEN":USING "##.#####^"
20 INPUT "C IN MIC.FARAD: ";C(N):C(N)=C(N)/1000:
   IF C(N)=0 THEN 40
30 CG=CG+ RCP C(N):N=N+1: GOTO 20
40 CG= RCP CG: PAUSE "C-GES. = ": PRINT CG
50 INPUT " U-GES. IN V : ";UG
60 F=CG*UG:N=1
70 PAUSE "U"; STR$ N;" = ": PRINT F/C(N)
80 N=N+1: IF C(N)<>0 THEN 70
90 PAUSE "W-GES. = ": PRINT .5*CG*UG^2
100 N=1
110 PAUSE "W"; STR$ N;" = ": PRINT .5*F^2/C(N)
120 N=N+1: IF C(N)<>0 THEN 110

```

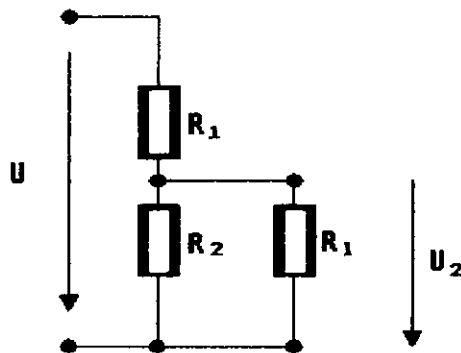
Belasteter Spannungsteiler

Beschreibung:

Dieses Programm errechnet die Spannung U_2 eines belasteten Spannungsteilers. Es müssen folgende Werte eingegeben werden:

U = Quellenspannung
 R1 = Oberer Widerstand
 R2 = Unterer Widerstand
 RL = Lastwiderstand (parallel zu R2)

SCHALTPLAN:



Listing:

```

2000 "D"
2005 PAUSE "BEL.SPANNUNGST."
2010 INPUT "R1=";R1 : INPUT "R2=";R2 : INPUT "RL=";RL :
      INPUT "U=";U
2015 X=R2/(R1+R2)
2020 U2=(U*X)/(1+((R1+R2)/RL)*(X-X^2))
2025 USING "&&&#.#####^"
2030 PRINT "U2=";U2 : USING : GOTO 2005
  
```

Serien-Parallel-Umwandlung

Beschreibung:

In der Elektronik ist es oftmals nötig, eine Reihenschaltung aus einem ohmschen Widerstand und einem Blindwiderstand in eine Parallelschaltung umzuwandeln, die aber bezüglich ihrer Eingangsklemmen den gleichen komplexen Widerstand aufweist. Mit diesem Programm ist es möglich eine Reihen- in eine Parallel- und eine Parallel- in eine Reihenschaltung umzuwandeln.

Funktion:

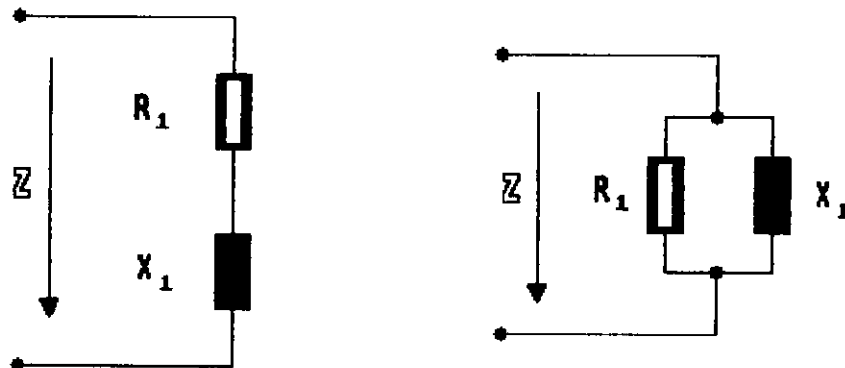
Nach dem Start erscheint im Display "RP ODER PR:", wobei der erste Buchstabe die momentane Schaltungsart und der zweite Buchstabe die Schaltungsart in die umgewandelt werden soll charakterisiert. Wollen wir nun eine Reihenschaltung in eine Parallelschaltung umwandeln, so geben wir "RP" «ENTER» ein. Nun fragt der Rechner nach Widerstand und Blindwiderstand, die wir dann eintippen müssen. Auf Tastendruck erscheint dann der ohmsche Parallelwiderstand und der dazu parallele Blindwiderstand.

Bei den Blindwiderständen kann es sich sowohl um kapazitive als auch um induktive Widerstände handeln, die sich folgendermaßen aus der Induktivität bzw. Kapazität berechnen:

$$X_{\text{ind.}} = \omega * L \quad X_{\text{kapa.}} = 1/(\omega * C)$$

Die aus dieser Rechnung erhaltenen Werte sind die jeweiligen Blindwiderstände.

SCHALTPLAN:



Listing:

```

10 INPUT " RP ODER PR: ";A$
20 IF A$<>"RP" AND A$<>"PR" THEN 10
30 GOTO A$
40 "PR" INPUT "R2= ";R2
50 INPUT "X2= ";X2
60 R1=R2/(1+(R2/X2)):X1=X2/(1+(X2/R2)^2)
70 PAUSE "  R1 =": PRINT R1: PAUSE "  X1 =": PRINT X1
80 GOTO 10
90 "RP" INPUT "R1=";R1
100 INPUT "X1=";X1
110 R2=R1*(1+(X1/R1)^2):X2=X1*(1+(R1/X1)^2)
120 PAUSE "  R2 =": PRINT R2: PAUSE "  X2 =": PRINT X2
130 GOTO 10

```

Schwingkreis

Beschreibung:

Diese Programm errechnet die Spule L, den Kondensator C oder die Frequenz F eines LC-Schwingkreises. Nach dem Start des Programmes fragt das Programm, welcher Wert errechnet werden soll und anschließend nach den beiden gegebenen Werten. Das Ergebnis wird danach einschließlich des Einheitswertes angezeigt.

Dabei bedeuten die Einheiten:

- P = Piko
- N = Nano
- U = Mikro
- M = Milli

Diese Vereinbarung ist aufgrund der fehlenden Kleinschrift des PC 1401/02 notwendig.

Listing:

```

7000 "A" : PAUSE "SCHWINGKREIS" : INPUT "C,L ODER F?";A$
7020 IF A$="C" THEN 7200
7030 IF A$="L" THEN 7130
7040 IF A$="F" THEN 7060
7050 BEEP 1 : PAUSE "FALSCH EINGABE" : GOTO 7000
7060 INPUT "C IN PF";C,"L IN UH";L
7070 F=1E9/(2*PI*SQR(C*L))
7080 IF F>1E6 THEN 7110
7090 IF F>1E3 THEN 7120
7100 PRINT "F=";F;" HZ" : GOTO 7000
7110 F=F/1E6 : PRINT "F=";F;" MHZ" : GOTO 7000
7120 F=F/1E3 : PRINT "F=";F;" KHZ" : GOTO 7000
7130 INPUT "C IN PF";C,"F IN HZ";F
7140 L=1E15/((2*PI*F)^2*C)
7150 IF L<1E-4 THEN 7180
7160 IF L<1 THEN 7190
7170 PRINT "L=";L;" MH" : GOTO 7000
7180 L=L*1E6 : PRINT "L=";L;" NH" : GOTO 7000
7190 L=L*1E3 : PRINT "L=";L;" UH" : GOTO 7000
7200 INPUT "F IN HZ";F,"L IN UH";L
7210 C=1E12/((2*PI*F)^2*L)
7220 IF C<1E-3 THEN 7250
7230 IF C<1 THEN 7260
7240 PRINT "C=";C;" UF" : GOTO 7000
7250 C=C*1E6 : PRINT "C=";C;" PF" : GOTO 7000
7260 C=C*1E3 : PRINT "C=";C;" NF" : GOTO 7000

```

Vitrometer

Beschreibung:

Programm zum auswählen von Widerständen mit Hilfe des Farbcodes. Es läßt sich ganz einfach der Ohmwert zu einem bestimmten Farbcode bestimmen, aber auch umgekehrt der Farbcode zu einem vorgegebenen Ohmwert.

```

Start : DEF "A" Farbcode -> Ohmwert
        "Z" Ohmwert -> Farbcode
        "S" E-Reihen

```

Listing:

```

100 "A" CLEAR: DIM A$(20)*10
110 A$(1)="SCHWARZ",A$(2)="BRAUN",A$(3)="ROT",
    A$(4)="ORANGE",A$(5)="GELB"
115 A$(6)="GRUEN",A$(7)="BLAU",A$(8)="VIOLETT",
    A$(9)="GRAU",A$(10)="WEIS"
117 A$(11)="GOLD",A$(12)="SILBER"

```

```

118 PRINT "FARBCODE-OHMWERT"
120 INPUT "1.RING :";X$, "2.RING :";Y$, "3.RING :
    ";W$, "4.RING :";V$
125 C=10
130 N=N+1
140 IF X$=A$(N) THEN LET B=B+(N-1)*C:GOTO 160
150 GOTO 130
160 M=M+1:N=0
170 IF M=1 THEN LET X$=Y$:C=1:GOTO 130
190 O=O+1
200 IF W$=A$(12) THEN LET B=B*0.01:GOTO 250
210 IF W$=A$(11) THEN LET B=B*0.1:GOTO 250
220 IF W$=A$(O) THEN LET B=B*10^(O-1):GOTO 250
230 GOTO 190
250 IF V$="BRAUN" THEN LET P=1:GOTO 300
260 IF V$="ROT" THEN LET P=2:GOTO 300
265 IF V$="GOLD" THEN LET P=5:GOTO 300
270 IF V$="GRUEN" THEN LET P=.5:GOTO 300
275 IF V$="SILBER" THEN LET P=10:GOTO 300
280 IF V$="BLAU" THEN LET P=.25:GOTO 300
285 IF V$="VIOLETT" THEN LET P=.1:GOTO 300
290 P=20
300 PRINT "OHMWERT : "
310 PRINT B;"+-";P;"%"
400 END
410 "Z" PRINT "OHMWERT-FARBCODE":CLEAR:DIM A$(20)*10
420 A$(1)="SCHWARZ",A$(2)="BRAUN",A$(3)="ROT",
    A$(4)="ORANGE",A$(5)="GELB"
430 A$(6)="GRUEN",A$(7)="BLAU",A$(8)="VIOLETT",
    A$(9)="GRAU",A$(10)="WEIS"
440 A$(11)="GOLD",A$(12)="SILBER"
510 INPUT "OHMWERT:";A
520 A=ABS A:N=0,M=0,C=0,P=0,B=0,D=A
524 IF A<10 THEN LET P=10
525 IF A<1 THEN LET P=11
528 GOTO 540
530 N=N+1
540 IF A>=10^N THEN LET B=A/10^N:C=A-(INT B*10^N)
545 IF B<10 THEN 600
550 GOTO 530
600 B=(INT B)+1:PRINT A$(B)
610 M=M+1
620 IF M>1 THEN 690
630 A=C:N=0:B=C:GOTO 530
690 D=(INT(D/1000))*100:IF P>0 THEN GOTO 750
695 GOTO 710
700 P=P+1
710 IF D=0 THEN GOTO 750
720 IF D>=10^P AND D<10^(P+1) THEN GOTO 750
730 GOTO 700
750 PRINT A$(P+1)
800 END
1000 "S" CLEAR:PRINT " E - REIHEN":INPUT " E";A
1010 PRINT A;"WERTE JE DEK."
1020 B=10^(1/A)
1030 PRINT "STUFENFAK:";B:D=.5
1040 IF A=6 LET D=20
1045 IF A=12 LET D=10
1050 IF A=24 LET D=5

```



```
1060 IF A=48 LET D=2
1065 IF A=96 LET D=1
1070 IF A=192 LET D=.5
1100 PRINT "+-";D;"%"
1110 INPUT "DEKADE:";E:PRINT "NUR NAEHERUNGEN"
1120 F=10*(E-1):IF F=0 THEN LET F=1
1125 PRINT F;" OHM"
1130 F=F*B:H=INT(F*100+0.5)/100
1140 PRINT H;" OHM"
1150 GOTO 1130
```

Kapitel 4.3

Maschinenbau

Festigkeitsberechnung

Beschreibung:

Dieses Programm berechnet die zulässige Spannung, die erforderliche Fläche oder bei Kreisflächen den erforderlichen Durchmesser und die elastische Verlängerung eines auf Zug beanspruchten Bauteils. Es darf sich dabei um runde oder quadratische Querschnitte handeln. Dazu müssen folgende Parameter eingegeben werden:

Werkstoffkennwert	R (Rm oder Reh bzw. Rp0,2)
Sicherheit	S (Sb oder Sf)
Kraft	F
E-Modul	E
Anfangslänge	L0

Listing:

```

6005 "F" PAUSE "FESTIGKEITSBER." :
    PAUSE "GES.ZUL.SPG.(SP)"
6010 PAUSE "UND ERF. FLAECHE" : PAUSE "WERKST.KENNWERT?"
6015 INPUT "R=";R : PAUSE "SICHERHEIT?"
6020 INPUT "S=";S : PAUSE "KRAFT?"
6025 INPUT "F=";F : PAUSE "E-MODUL?" : INPUT "E=";E
6030 PAUSE "ANFANGSLAENGE?" : PAUSE "IN MM" :
    INPUT "L0=";L0
6035 SP=R/S : A=F/SP
6040 D=SQR((4*A)/PI) : L=(SP/E)*L0
6045 PRINT "SP=";SP : PRINT "A=";A
6050 PAUSE "D BEI A-KREISFL." : PRINT "D=";D
6055 PAUSE "EL. VERLAENGERUNG" : PRINT "L=";L
6060 GOTO 6005

```

Momentensatz

Beschreibung:

Das Programm berechnet nach der Eingabe aller bekannten Kräfte und Längen, an einem drehbar gelagerten Balken (z.B. eine Wippe), die noch unbekannte Kraft bzw. Länge. Das Programm ist auf dem Momentengleichgewichtssatz aufgebaut, der besagt, daß im Gleichgewichtszustand die Summe aller am Körper von außen angreifenden Momente gleich Null ist.

Funktionsweise:

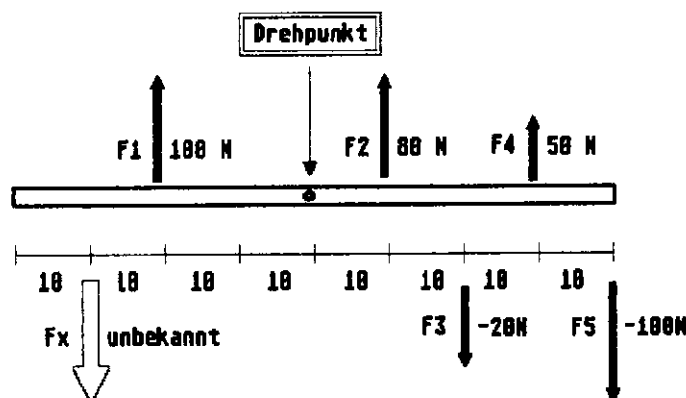
Nach dem Programmstart und der Anzeige "MOMENTENSATZ", werden Sie gefragt: "F IN NEWTON:". Geben Sie nun die erste Kraft in Newton ein. Zeigt die Kraft nach oben, bekommt sie ein positives, zeigt sie nach unten bekommt sie ein negatives Vorzeichen. Danach erscheint "L IN MILLIM.:". Tippen Sie nun den Abstand der ersten Kraft in mm von der Drehachse ein. Wirkt die Kraft rechts von der Drehachse, wird die Länge positiv, wirkt sie links von der Drehachse, wird die Länge negativ eingegeben.

Hat man nun alle bekannten Kräfte und die dazugehörigen bekannten Längen eingegeben, tippt man nach "F IN NEWTON:" einfach «0» «ENTER». Da man meist eine unbekannte Kraft mit bekannter Länge oder eine unbekannte Länge mit bekannter Kraft berechnen muß, werden Sie gefragt, was Sie berechnen möchten - es blinkt "BERECHNEN? «F/L»". Haben Sie eine unbekannte Kraft mit bekannter Länge zu berechnen, so drücken Sie für ca. 1 Sek. «F». Nun werden Sie gebeten, die ja bekannte Länge, einzutippen. Nach «ENTER» erscheint die Kraft in Newton im Display.

Beispiel:

Rechner:	Eingabe:
"MOMENTENSATZ"	
"F IN NEWTON:"	-100 «ENTER»
"L IN MILLIM.:"	40 «ENTER»
"F IN NEWTON:"	50 «ENTER»
"L IN MILLIM.:"	30 «ENTER»
"F IN NEWTON:"	-20 «ENTER»
"L IN MILLIM.:"	20 «ENTER»
"F IN NEWTON:"	80 «ENTER»
"L IN MILLIM.:"	10 «ENTER»
"F IN NEWTON:"	100 «ENTER»
"L IN MILLIM.:"	-20 «ENTER»
"F IN NEWTON:"	0 «ENTER»
"BERECHNEN? «F/L»"	«F»
"L IN MILLIM.:"	-30
"F IN NEWTON:"	
"-1.36666666E 02"	

Ein negatives Vorzeichen sagt aus, daß die Kraft nach unten wirkt.

Listing:

```

10 PAUSE " MOMENTENSATZ"
20 M=0
30 F=0: INPUT "F IN NEWTON: ";F: IF F=0 THEN 60
40 L=0: INPUT "L IN MILLIM. : ";L:M=M+F*L
50 GOTO 30
60 PAUSE "BERECHNEN? <F/L>":A$= INKEY$ : IF
  A$<>"F" AND A$<>"L" THEN 60
70 IF A$="F" THEN 90
80 F=0: INPUT "F IN NEWTON: ";F:L=M/F*(-1): PAUSE
  " LAENGE IN MM:": USING "##.#####^": PRINT L:
  USING : GOTO 20
90 INPUT "L IN MILLIM. :";L:F=M/L*(-1): PAUSE " F IN NEWTON:":
  USING "##.#####^": PRINT F: USING : GOTO 20

```

Riemengetriebe

Beschreibung:

In der Mechanik stößt man oft auf Probleme, bei denen eine Scheibe über einen Riemen angetrieben wird.

Oft sind dabei drei der vier benötigten Werte N1 (Drehzahl treibende Scheibe), D1 (Durchmesser treibende Scheibe), N2 (Drehzahl getriebene Scheibe) und D2 (Durchmesser getriebene Scheibe) gegeben.

Dieses Programm fragt Sie nun nach allen vier Werten ab, wobei Sie bei der Abfrage des Ihnen unbekanntes Wertes nur «ENTER» drücken. Vom Rechner wird automatisch die für die Berechnung benötigte Formel ausgewählt, der gesuchte Wert berechnet und angezeigt.

Die Werte der Drehzahlen müssen in 1/min und die der Durchmesser in mm angegeben werden.

Weiterhin berechnet das Programm die Riemengeschwindigkeit in m/s, das Übersetzungsverhältnis und ob die getriebene Scheibe langsamer oder schneller als die treibende Scheibe ist.

Funktionsweise:

Wird bei den SHARP-PC's bei einer Eingabe nur «ENTER» gedrückt, so bricht der Rechner die aktuelle Zeile ab und springt in die darauffolgende Zeile. Dieser Effekt wurde hier folgendermaßen ausgenutzt: Gibt man eine Zahl «ENTER» ein, so weißt der Rechner der Variablen die Zahl zu und führt den darauffolgenden Sprung aus. Gibt man aber bei der gesuchten Angabe nur «ENTER» ein, wird die Zeile abgebrochen und somit wird der GOTO-Befehl (der die nächste Zeile überspringt) nicht ausgeführt. Deshalb kommt der Rechner in die Zeile, in der der Variablen der Name der unbekanntes Größe zugeordnet wird. Da dies normalerweise nur einmal geschieht, muß A\$ 2-stellig sein (sonst Fehler!).

Nach der Überprüfung von A\$ wird in das entsprechende Unterprogramm verzweigt, welches den gewünschten Wert berechnet. Die Variable B dient ausschließlich zur Ausgabe, damit nicht 4 verschiedene Ausgaberroutinen geschrieben werden müssen.

Beispiel:

Rechner:	Eingabe:	
RIEMENGETRIEBE		«ENTER»
D1 =	2	«ENTER»
N1 =	3	«ENTER»
D2 =		«ENTER»
N2 =	6	«ENTER»
D2 =		«ENTER» (berechneter Wert)
1		«ENTER»
GESCHWINDIGKEIT=		«ENTER»
3.141592654E-04		«ENTER»
UEBERSETZUNG=		«ENTER»
0.5		«ENTER»
INS SCHNELLE		«ENTER»

Listing:

```

20 CLEAR: PRINT "RIEMENGETRIEBE"
30 INPUT "D1 = ";D1: GOTO 50
40 A$=A$+"D1"
50 INPUT "N1 = ";N1: GOTO 70

```

```

60 A$=A$+"N1"
70 INPUT "D2 = ";D2: GOTO 90
80 A$=A$+"D2"
90 INPUT "N2 = ";N2: GOTO 110
100 A$=A$+"N2"
110 IF LEN A$<>2 THEN 20
120 GOSUB A$
130 PRINT A$;" =": PRINT B
140 V=( PI *D1*N1)/60000
150 PRINT "GESCHWINDIGKEIT=": PRINT V
160 I=N1/N2
170 PRINT "UEBERSETZUNG =": PRINT I
180 IF I>1 THEN PRINT "INS LANGSAME"
190 IF I<1 THEN PRINT "INS SCHNELLE"
200 GOTO 10
210 "D1"D1=(N2*D2)/N1:B=D1: RETURN
220 "N1"N1=(N2*D2)/D1:B=N1: RETURN
230 "D2"D2=(N1*D1)/N2:B=D2: RETURN
240 "N2"N2=(N1*D1)/D2:B=N2: RETURN

```

Brinell-Härte

Beschreibung:

Dieses Programm dient zum Errechnen der Brinellhärte aus den Werten der Härteprüfung. Dadurch werden die Tabellen zum Nachschlagen der Härtegrade überflüssig. Die Angabe von Kugeldurchmesser und Eindruckdurchmesser erfolgt in mm, die Angabe der Prüfkraft in Newton. In Zeile 6060 wird abgefragt, ob Prüfkraft oder Kugeldurchmesser geändert werden sollen. Wird "J" eingegeben, so können diese neu eingegeben werden, wird nur "ENTER" gedrückt, wird mit den alten Werten weitergerechnet. Der Start des Programmes erfolgt mit 'DEF B'.

Listing:

```

6010 "B" PAUSE "BRINELLHAERTE"
6020 INPUT "KUGELDM.=";K : INPUT "PRUEFKRAFT/N=";F
6030 INPUT "EINDRUCK/MM=";D
6040 H=(.102*2*F)/(PI*K*(K-SQR(K^2-D^2)))
6050 USING "&&&####.##" : PRINT "HB=";H : A$="" : USING
6060 INPUT "F OD. K AENDERN";A$
6070 IF A$="J" THEN 6010
6080 GOTO 6030

```

Statik

Beschreibung:

Dieses Programm wurde aufgrund des Statik-Unterrichtes in der 12.Klasse der Fachoberschule auf einem PC-1401 geschrieben. Es berechnet ein allgemeines Kräftesystem mit einem zweiwertigen Lager (gleichzeitig bezogenem Null- und Drehpunkt), einem einwertigen Lager (von dem die Koordinaten und der Winkel der Wirkungslinie bekannt sein müssen) und bis zu 20 Kräften (max. Speicher beim 1401)

Das Programm erwartet die Daten ab Zeile 1000 als DATA-Zeilen. Als erstes wird die Anzahl der Kräfte angegeben.

Dann kommen die Kräfte selber. Als erstes wird die Stärke der Kraft angegeben, dann der Winkel bezogen auf den mathematischen Einheitskreis (gegen Uhrzeigersinn) und als letztes die Koordinaten des Angriffspunktes. Nach der in der ersten Zeile angegebenen Anzahl von Kräften kommen die Daten für das einwertige (B) Lager.

Es können auch mehrere Kräftesysteme eingegeben werden. Diese müssen dann aber immer mit Tausenderzeilen beginnen.

Beispiel:

1. SYSTEM

1000	DATA 1	Anzahl der Kräfte 1
1010	DATA 10,280,3,2	1.Kraft mit 10kN, 280 , 3m X und 2m Y-Koordinate
1020	DATA 12,0,90	B-Lager 12m X und auf der X-Achse

2. SYSTEM

2000	DATA 3	Anzahl der Kräfte 3
2010	DATA 20,280,3,2	1. Kraft
2020	DATA 30,270,6,4	2. Kraft
2030	DATA 35,300,9,2	3. Kraft
2040	DATA 12,0,70	B-Lager (einwertiges)

Werden die Daten für das B-Lager weggelassen gibt es entweder einen Error vor dem berechnen der Auflagerkräfte (READ ohne DATA) oder irgendwelche Ergebnisse für die Auflagerkräfte die dann natürlich zu ignorieren sind.

Jetzt kann das Programm mit DEF 'S' gestartet werden. Als erstes muß die Nummer des Kräftesystemes angegeben werden. Danach wird zur Bestätigung noch einmal die Anzahl der Kräfte angezeigt. Jetzt ist fast nur noch mit ENTER fortzufahren. Es werden im weiteren Verlauf des Programmes angezeigt:

TEILKRAEFTE?
 RESULTIERENDE?
 MOMENTE?
 AUFLAGERKRAEFTE?

Hierauf kann entweder mit ',' oder 'L' und anschliessendem ENTER geantwortet werden.

Mit ',' werden die entsprechenden Ergebnisse übersprungen und mit 'L' werden sie angezeigt.

Im Wiederholungsfall braucht nur ENTER gedrückt zu werden.

Die Ergebnisse werden aber in jedem Fall berechnet und sind auch später noch im RUN Modus abrufbar (nur für das zuletzt berechnete System).

Folgende Variablen wurden benutzt:

Feld F(x,y)	'x'=1	Nr. der Kraft	'y'=0	Kraft
	.	.	.	1 Winkel
	.	.	.	2 X-Koordinate
	.	20	.	3 Y-Koordinate
	.	.	.	4 X Teilkraft
	.	.	.	5 Y Teilkraft
	.	.	.	6 X Moment
	.	.	.	7 Y Moment

'FR' Kraft der Resultierenden mit 'RX' + 'RY' Teil
 'AR' Winkel der Resultierenden bezogen auf Einheitskreis
 (beim Ablauf wird mit AR zuerst der 'Taschenrechnerwert'
 angezeigt, dann mit AR EHK der Wert auf Einheitskreis bezogen)
 'MR' Resultierendes Drehmoment
 'L' Hebelarm der Resultierenden
 'FB' Auflagerkraft für das einwertige B-Lager mit 'CX' + 'CY' Teil
 (beim Ablauf wird BX und BY angezeigt)
 'BA' Richtiger Winkel für B-Lager (im Ablauf AB)
 'FA' Kraft von Auflagerkraft des zweiwertigen A-Lagers mit
 'AX' und 'AY' als Teilkräfte
 'AA' Winkel der FA Kraft, hier gilt gleiches wie bei 'AR'

Listing:

```

100 "S":CLEAR:INPUT "DATENSATZ";R
110 RESTORE 1000*R:READ K:PRINT K;" KRAEFTE"
120 DIM F(K,7)
130 FOR R=1 TO K
140 FOR T=0 TO 3
150 READ F(R,T)
160 NEXT T:NEXT R
170 INPUT "TEILKRAEFTE?";Z$:GOSUB 900
180 FOR R=1 TO K
190 F(R,4)=F(R,0)*COS(F(R,1)):PRINT "F";R;"X";F(R,4)
200 F(R,5)=F(R,0)*SIN(F(R,1)):PRINT "F";R;"Y";F(R,5)
210 NEXT R
220 INPUT "RESULTIERENDE?";Z$:GOSUB 900
230 FOR R=1 TO K
240 RX=RX+F(R,4):RY=RY+F(R,5)
250 NEXT R:PRINT "RX";RX:PRINT "RY";RY
260 IF RX=0 THEN LET AR=90+180*(RY<0):GOTO 300
270 AR=ATN(RY/RX):PRINT "AR";AR
280 IF RX<0 THEN LET AR=180+AR:GOTO 300
290 IF AR<0 THEN LET AR=AR+360
300 PRINT "AR EHK";AR
310 FR=SQR(RX^2+RY^2):PRINT "FR";FR
320 INPUT "MOMENTE?";Z$:GOSUB 900
330 FOR R=1 TO K
340 F(R,6)=F(R,3)*F(R,4):MX=MX+F(R,6):
   PRINT "M";R;"X";F(R,6)
350 F(R,7)=-F(R,5)*F(R,2):MY=MY+F(R,7):
   PRINT "M";R;"Y";F(R,7)
360 NEXT R
370 MR=MX+MY
380 IF MR=0 THEN 950
390 PRINT "MX";MX:PRINT "MY";MY

```



```
400 PRINT "MR";MR
410 IF RY=0 THEN LET L=ABS(MR)/RX:PRINT "LX";L:GOTO 450
420 L=ABS(MR)/RY
430 IF RY<0 THEN LET L=-L
440 PRINT "L";L
450 INPUT "AUFLAGERKRAEFTE";Z$:GOSUB 900
460 READ BX, BY, BA
470 DIM LB(1),MB(1),FB(1),CX(1),CY(1)
480 LB(0)=ABS(SIN(BA)*BX)+ABS(COS(BA)*BY)
490 LB(1)=ABS(SIN(BA)*BX)-ABS(COS(BA)*BY)
500 FOR R=0 TO 1
510 FB(R)=ABS(MR/LB(R))
520 CX(R)=FB(R)*COS(BA)
530 CY(R)=FB(R)*SIN(BA)
540 MB(R)=CX(R)*BY-CY(R)*BX
550 NEXT R
560 IF ABS(ABS(MR)-ABS(MB(0)))>ABS(ABS(MR)-ABS(MB(1)))
    THEN 580
570 FB=FB(0):MB=MB(0):CX=CX(0):CY=CY(0):LB=LB(0):
    GOTO 590
580 FB=FB(1):MB=MB(1):CX=CX(1):CY=CY(1):LB=LB(1)
590 LB=ABS(LB):IF LB=0 THEN 950
600 IF ABS(MR-MB)>ABS(MR+MB) THEN 620
610 BA=BA+180-360*(BA>=180):CX=-CX:CY=-CY
620 PRINT "BX";CX:PRINT "BY";CY:PRINT "FB";FB
630 PRINT "AB";BA
640 AX=- (RX+CX)
650 AY=- (RY+CY)
660 FA=SQR(AX^2+AY^2)
670 IF FA=0 THEN 950
680 PRINT "AX";AX:PRINT "AY";AY
690 PRINT "FA";FA
700 IF AX=0 THEN LET AA=90+180*(AY<0):GOTO 740
710 AA=ATN(AY/AX):PRINT "AA";AA
720 IF AX<0 THEN LET AA=AA+180:GOTO 740
730 IF AA<0 THEN LET AA=AA+360
740 PRINT "AA EHK";AA
899 END
900 IF Z$="," THEN WAIT 0:GOTO 920
910 WAIT
920 RETURN
950 WAIT:PRINT "KEIN ALLGEMEINES"
960 END
1000 DATA 1
1010 DATA 10,280,3,2
1020 DATA 12,0,90
2000 DATA 3
2010 DATA 20,280,3,2
2020 DATA 30,270,6,-4
2030 DATA 35,300,9,2
2040 DATA -2,-3.5,70
```

Sparrenberechnung

Beschreibung:

Dieses Programm ist für Architekten, Bauingenieure oder Zimmermänner (-frauen) gedacht, die die Dimension für Dachsparren etwas genauer als mit den üblichen Faustformeln vornehmen wollen. Das Programm ist auf verschiedene, auf den "Normalfall" bezogene, Grundlagen aufgebaut. Die Gesamtbelastung wird mit $Q=1,4\text{KN/m}$ angenommen. Der Sparren wird als Einfeldträger berechnet.

DER PROGRAMMABLAUF

Die erste verlangte Eingabe ist die Sparrenlänge in der Grundrißprojektion. Anschließend wird der wahre Winkel des Sparrens zur Horizontalen erfragt. Ist dieser nicht bekannt, muß "0" eingegeben werden und das Programm fragt nach der Höhendifferenz zwischen Sparrenfuß und oberem Auflager.

Jetzt berechnet der PC Widerstands- und Trägheitsmoment, sowie die wahre Sparrenlänge und gibt diese aus. Nun wählt der PC Sparrenquerschnitte deren Seitenverhältnis $b/h \geq 0.6$ ist aus und zeigt diese an.

Letzte Programmstufe ist die Berechnung des Holzverbrauches und die Anzeige dieses Wertes.

Listing:

```

3000 "S"
3010 PAUSE "SPARRENDIMENSION"
3020 PAUSE "BEI SPARRENBELAST."
3030 PAUSE "Q=1.4KN/M"
3040 PAUSE "EINGABE D.MASZE"
3050 PAUSE "IN GR.PROJEKTION"
3055 WAIT 160
3056 CLEAR
3060 INPUT "SP.LAENGE L=" ; L
3061 INPUT "WINKEL=" ; W
3062 S=L/COS W
3063 IF S=L GOSUB 3455
3070 LET M=(1.4*SQU L)/8
3080 A=(1.4*L)/2
3090 EW=M*100
3100 EI=208*M*S
3101 PRINT "ERF W=" ; EW ; "CM3"
3102 PRINT "ERF I=" ; INT EI ; "CM4"
3104 PRINT "SP.LAE.=" ; S
3105 IF EW>1344 OR EI>16128 GOSUB 3435
3114 FOR B=6 TO 14 STEP 2
3115 FOR HO=8 TO 24 STEP 2
3130 WI=(B*HO*HO)/6
3140 I=(B*(HO)^3)/12
3145 LET V=B*HO*S*10
3150 IF WI>=EW AND I>=EI-(EI*0.04)
      AND B/HO<=0.6 GOSUB 3400 3160 NEXT HO
3170 NEXT B
3180 PAUSE "DAS WARS!"
3200 INPUT "WEITER ? J/N " ; J$
3210 IF J$="J" OR J$="JA" GOTO 3056

```

```
3220 IF J$="N" OR J$="NEIN" GOTO 3460
3400 WAIT 160
3405 IF WI>EW+(EW*0.1) AND I>EI+(EI*0.1) GOTO 3430
3406 BEEP 1
3410 PRINT "B/H= ";B;"/";HO
3420 PRINT "HOLZVERB. ";V;"DM3"
3430 RETURN
3435 BEEP 2
3440 PAUSE "UNWIRTSCHAFTLICH":GOTO 3056
3450 RETURN
3455 INPUT "HOEHE=";H
3456 LET S=SQR(SQU L+SQU H)
3457 RETURN
3460 PAUSE "TSCHUESS!!!!!!!!!"
3470 END
```

Kapitel 4.4

Spiel & Spaß

Reaktionstest

Beschreibung:

Dieses Programm zeigt, daß man mit dem Sharp-PC durchaus auch zeitkritische Probleme lösen kann.

Funktionsweise:

Starten des Programms mit DEF A. Nach Anzeige von "ACHTUNG" ist für einige Sekunden ein schwarzes Rechteck sichtbar. Wenn es verschwindet, dann so schnell wie möglich die BREAK-Taste drücken. Nun DEF Z drücken, es erfolgt die Auswertung und Anzeige der Reaktionszeit in Millisekunden. (Toleranz +-20 ms)

Listing:

```

810 REM REAKTIONSTEST
820 "Z" D=C*50: GOSUB 1000
830 D=D+V: WAIT 200: PRINT D;"MILLISEC."
840 WAIT 150: PRINT "REAKTION:"
850 IF D<170 GOTO 880
855 IF D<220 GOTO 885
860 IF D<270 GOTO 890
865 IF D<380 GOTO 895
870 GOTO 900
880 PRINT "(( SEHR GUT ! ))": GOTO 910
885 PRINT "((( GUT )))": GOTO 910
890 PRINT "$ NOCH NORMAL $": GOTO 910
895 PRINT "=== SCHWACH ===": GOTO 910
900 WAIT 250: PRINT " LAHMER SOCKEN !"
910 IF D>F GOTO 930
920 F=D: BEEP 4: PRINT "XX REKORD ! XX"
930 PRINT "REK.:";F;"M.SEC."
935 C=0: WAIT 100: PRINT " NOCH MAL ?": WAIT :
PRINT " TASTE (ENTER)": GOTO 950
940 "A" CLEAR :F=230: WAIT 200: PRINT "-REAKTIONSTEST-"
950 WAIT 150: PRINT "((( ACHTUNG )))"
960 GOSUB 1000: WAIT 159+V: PRINT " **** ":
IF INKEY$ ="S" THEN 986
970 FOR C=1 TO 299: IF INKEY$ ="S" THEN 810
973 NEXT C
980 WAIT : PRINT "KEINE REAKTION ?"
986 WAIT 150: PRINT "! BESCHUMMELT !":
PRINT " ODER NERVOES ?"
990 END
1000 RANDOM :V= RND 23: RETURN

```

Frogger

Beschreibung:

Dies ist das bekannte Spiel FROGGER in seiner Version für die SHARP-PC's. Das Programm sollte möglichst in den leeren Speicher des Rechners eingegeben werden, da andere Programme unter Umständen vom Unterprogramm zum Erzeugen der Autos zerstört werden können.

Ist die Initialisierung beendet, d.h. die Autos erzeugt, so kann das Programm problemlos auf Cassette gespeichert oder in andere Rechner übertragen werden. Es können dann auch zusätzliche Programme eingetippt werden. Nach dem Starten des Programms wird der letzte Hi-Score angezeigt. Um nun das eigentliche Spiel zu beginnen, muß für ca. 2 Sekunden die Taste "F" gedrückt werden. Es erscheint "ACHTUNG" und das Spiel beginnt.

Spielverlauf:

Ihr Frosch muß eine sehr verkehrsreiche Straße überqueren, wobei er nur auf dem Fußgängerüberweg sicher ist. Wird eine Lücke im Verkehr frei, kann der Frosch auf die Straße springen. Er darf jedoch nie auf der Fahrbahn sitzen bleiben, sondern muß immer hüpfen. Vorsicht auch bei Fußgängern und Schlangen auf dem Fußweg, da der Frosch ebenfalls stirbt, wenn er auf sie springt. Ist man auf dem mittleren Weg, so beginnt die 2. Runde. Hier muß der Frosch einen Fluß überqueren. Um zum Laichplatz zu kommen, kann der Frosch auf Schildkröten und Baumstämme hüpfen. Auf Baumstämmen ist er sicher, bis sich ein Krokodil nähert. Auf Schildkröten kann er nicht lange sitzen, da diese wegtauchen können. Dauert die Zeit um zum Laichplatz zu gelangen zu lange, so stirbt der Frosch. Je Spiel gibt es 4 Frösche. 100 Punkte für jeden Frosch im Laichplatz, 1 Extrafrosch für 400 Punkte.

Steuerung: Taste "V" -> Vorwärts
Taste "Z" -> Zurück

Die Tasten immer mindestens 1 Sekunde drücken !

Figuren:	X	Frosch	Ⓢ	Schlange
	!	Baum oder Fußgänger	*	Schildkröte
	:■:	Auto	:=	Krokodil
	I	Fußweg)	Laichplatz

Listing:

```

5   "F" CLEAR: BEEP 1
10  PAUSE "%% FROGGER %%"
11  BEEP (1): PAUSE "** HI-SCORE **": BEEP (1): PAUSE
    "-> ";Q: IF INKEY$ ="F" THEN 13
12  GOTO 10
13  B=4:C=40:D=0:E=0: PAUSE "ACHTUNG": PAUSE "ACHTUNG"
14  GOSUB 400: IF V=2 THEN 20
15  PAUSE "X:?:?:?:?:I"
16  IF INKEY$ ="V" THEN 90
17  GOTO 14
20  PAUSE "X :?:?:?:?:I"
22  IF INKEY$ ="V" THEN 25
23  GOTO 14
25  GOSUB 300: GOSUB 400: IF V=2 THEN 30
27  PAUSE "I X :?:?:?:I"
28  IF INKEY$ ="Z" THEN 14
29  GOTO 90
30  PAUSE "! X :?:?:?:I"
32  IF INKEY$ ="V" THEN 35
33  GOTO 90
35  GOSUB 300: GOSUB 400: IF V=2 THEN 40
36  PAUSE "I X :?:?:?:I"
37  IF INKEY$ ="Z" THEN 25
38  GOTO 90

```

```

40 PAUSE "I:?: X I"
42 IF INKEY$="V" THEN 45
43 GOTO 90
45 GOSUB 300: GOSUB 400: IF V=2 THEN 55
48 PAUSE "I:?: X §"
50 IF INKEY$="Z" THEN 35
52 GOTO 90
55 PAUSE "!:?:?: X I"
57 IF INKEY$="V" THEN 60
58 GOTO 90
60 PAUSE "!:?:?:?:X": GOTO 99
68 PAUSE "!:?:?:?:?:X": GOTO 99
90 GOSUB 500: GOTO 14
99 BEEP 1
100 A=2: GOSUB 400: IF V=2 THEN 108
102 PAUSE "X * * )"
104 IF INKEY$="V" THEN 111
106 GOTO 100
108 PAUSE "X !!!* )": IF INKEY$="V" THEN 112
110 IF INKEY$="" THEN 100
111 GOSUB 500: GOTO 100
112 GOSUB 300: GOSUB 400: IF V=2 THEN 118
114 PAUSE "I X=: )": IF INKEY$="Z" THEN 100
116 GOSUB 500: GOTO 100
118 PAUSE "§ X* * )": IF INKEY$="V" THEN 126
120 IF INKEY$="" THEN 112
124 GOSUB 500: GOTO 100
126 GOSUB 300: GOSUB 400: IF V=2 THEN 132
128 PAUSE "I !X* )": IF INKEY$="Z" THEN 112
130 GOSUB 500: GOTO 100
132 PAUSE "I X!* )": IF INKEY$="V" THEN 136
134 GOSUB 500: GOTO 100
136 GOSUB 300: GOSUB 400: IF V=2 THEN 142
138 PAUSE "I *X=: )": IF INKEY$="V" THEN 148
142 PAUSE "I ! X* )": IF INKEY$="V" THEN 148
144 IF INKEY$="" THEN 136
146 GOSUB 500: GOTO 102
148 GOSUB 300: GOSUB 400: PAUSE "I !* X )"
150 IF INKEY$="V" GOSUB 154
152 GOSUB 500: GOTO 100
154 GOSUB 168: GOTO 14
160 PAUSE "I !!!* X": GOSUB 700: E=E+100+G: D=D+1: P=D-4:
IF P=0 GOTO 166
164 GOTO 168
166 B=B+1: D=0: BEEP 2: PAUSE "+ 1 X"
168 C=31: GOSUB 200: IF B=0 GOTO 503
170 RETURN
200 WAIT 122: PRINT "---";E;" PUNKTE---": RETURN
300 C=C-1: IF C=0 GOTO 600
301 RETURN
400 RANDOM :V= RND 2: RETURN
500 B=B-1: PAUSE "TOT!": PAUSE B;"X"
501 GOSUB 700: E=E+G: IF B<>0 GOTO 504
503 FOR I=1 TO 4: GOSUB 200: NEXT I: GOTO 510
504 C=40: RETURN
510 IF E>Q THEN LET Q=E
520 GOTO 10
600 GOSUB 500: IF A=2 GOTO 100
602 GOTO 14
700 G=(40-C)*5: RETURN

```

Nach der Eingabe des Programms, muß folgendes Hilfsprogramm eingegeben und dann einmal gestartet werden - danach kann es wieder gelöscht werden (es erzeugt die Autos im Display).

```
900 "A": FOR I=8193 TO 9193
910 IF(PEEK I)=63 GOTO 940
920 NEXT I : BEEP1
930 END
940 POKE I,254:NEXT
```

Buchstaben Erraten

Beschreibung:

Dies Programm ist, wie der Name schon vermuten läßt, ein Spielprogramm. Bei dem Spiel gilt es, einen Ihnen unbekanntem Buchstaben zu erraten. Sie drücken für ca. 1 Sek. auf die Buchstabetaste Ihrer Wahl, ist Ihr Buchstabe "größer" (A ist der "kleinste" Buchstabe) als der zu erratende Buchstabe, so erscheint er im rechten Teil, ist er "kleiner", so wird er im linken Teil des Displays angezeigt. So können Sie sich immer näher an den zu erratenden Buchstaben herantasten. Wurde der richtige Buchstabe eingegeben, zeigt Ihnen der Rechner die Anzahl der benötigten Versuche an.

Funktionsweise:

Nach dem Programmstart blinkt "A < \$ > Z". Das \$ soll den unbekanntem Buchstaben darstellen. Nehmen wir einmal an, es handle sich dabei um das "M". Drücken Sie nun während des Blinkvorgangs für ca. 1 Sek. die Taste <D>, so erscheint "D < \$ > Z". Nun wissen Sie, daß sich der gesuchte Buchstabe zwischen "D" und "Z" befindet. Drückt man jetzt <O>, so blinkt "D < \$ > O". Dies sagt Ihnen wiederum, der gesuchte Buchstabe befindet sich zwischen "D" und "O".

Das Spiel wurde so programmiert, daß wenn Sie jetzt z.B. auf <R> drücken sollten, das <R> ignoriert wird, da Sie mit <O> schon näher am zu erratenden Buchstaben waren. Dies verhindert eine höhere Zahl von Versuchen, durch einen unbeabsichtigten Tastendruck.

Drücken Sie nun auf <M> piepst der Rechner und zeigt Ihnen "RICHTIG", sowie die Anzahl der Versuche an. Das Spiel kann auch mit mehreren Personen gespielt werden indem die Versuche der verschiedenen Personen notiert und am Ende zusammengezählt werden. Derjenige, der die wenigsten Versuche hat, hat gewonnen.

<u>Beispiel:</u>	Rechner:	Eingabe:
	"** BUCHSTABEN **"	
	"*** ERRATEN ***"	
	"A < \$ > Z"	<K>
	(BEEP)	
	"K < \$ > Z"	<S>
	(BEEP)	
	"K < \$ > S"	<W> (wird ignoriert!)
	"K < \$ > S"	<O>
	(BEEP)	


```

"K < $ > O"          <M>
(BEEP)
"M < $ > O"          <N>
(BEEP)/(BEEP)...
" 5 VERSUCHE"
"A < $ > Z"
usw.

```

Listing:

```

10 PAUSE "## BUCHSTABEN ##": PAUSE "### ERRATEN ###":
RANDOM
20 Z= RND 89: IF Z<66 THEN 20
30 A$= CHR$( 65):B$= CHR$( Z):C$= CHR$( 90):V=0:D$="":
WAIT 15
40 PRINT " ";A$;" < $ > ";C$:D$= INKEY$ : IF D$<=A$ OR
D$> = C$ THEN 40
50 V=V+1: BEEP 1
60 IF D$<B$ THEN LET A$=D$
70 IF D$>B$ THEN LET C$=D$
80 IF D$=B$ AND V=1 THEN PAUSE " *** SUPER ***": WAIT 0:
FOR N=1 TO 15: CALL 1444: PRINT "": NEXT N: GOTO 20
90 IF D$=B$ THEN PAUSE " *** RICHTIG ***": WAIT 1: FOR
N=1 TO 5: CALL 1440: PRINT "": NEXT N: PAUSE " ";
STR$( V);" VERSUCHE": GOTO 20
100 GOTO 40

```

Ghostbuster

Beschreibung:

Dies ist das Spiel zu dem bekannten Kinofilm "GHOSTBUSTERS" in seiner Version für den PC 1401/02 mit zahlreichen Soundeffekten. Dabei wird von folgender Situation ausgegangen: Man befindet sich im Speisesaal eines grossen Hotels und muß die dort umherschwirrenden Geister fangen. Da die Geister nicht sichtbar sind, müssen sie mit einem Peilgerät geortet werden. Leider benötigt dieses Gerät sehr viel Strom, so daß die Energie sehr schnell abnimmt. Gelingt es aber einen Geist zu fangen, so kann man ihn ausquetschen und die gewonnene Energie seinem Ortungsgerät zuführen. Ziel des Spieles ist es nun, möglichst viele Geister zu fangen, bevor die Energie verbraucht ist.

Spielverlauf:

Das Spiel wird mit RUN gestartet und es erscheint abwechselnd der Titel und das höchste Spielergebnis. Wird nun für ca. eine Sekunde die «G»-Taste gedrückt, so beginnt das Spiel.

Man befindet sich dabei in einem Koordinatenfeld von 10×10 Feldern. Auf der Anzeige des Rechners werden die Koordinaten der eigenen Position, der Abstand zum Geist und die noch zur Verfügung stehende Energie angezeigt. Das Spiel ist beendet, sobald alle Energie verbraucht ist.

Wenn die Entfernung zum Geist gering genug ist, so hört man ihn piepsen. Wenn die Entfernung drei Einheiten beträgt, piepst der Rechner einmal, bei zwei Einheiten zweimal und bei einer Einheit dreimal. Wenn der Geist über den Spieler hinwegfliegt, so hört man ein Schwirren

Spielsteuerung:

Mit den Zahlentasten 2,4,6,8 steuert man die Richtung, in die man sich bewegen will. Dabei haben die Tasten folgende Bedeutung:

```

          oben
          8
links 4          6 rechts
          2
          unten

```

Die Tasten müssen für jeweils eine Sekunde gedrückt werden. Die Energie die man aus einem gefangenen Geist ausquetschen kann ist je nach Sorte des Geistes verschieden. Welche Sorte Geist man jedoch gerade verfolgt, ist nicht feststellbar.

Listing:

```

2 POKE &45E8,&00,&FF,&4F,&12,&5F,&61,&10,&DF,&60,&EF,
  &DF,&41,&7F,&45,&F9,&2D,&0E,&37
4 "G": RANDOM
5 WAIT 1: PRINT "##GHOST-BUSTER##": CALL 1442:
  GOSUB 700:G$=INKEY$: IF G$="G" THEN 20
10 PAUSE "** HIGH-SCORE **": WAIT 0: PRINT " >>> ";
  STR$(T);" <<<": CALL 1442: CALL &45E8:
  CALL &45E8:G$= INKEY$
15 IF G$<>"G" THEN 5
20 X=0:Y=0:E=50:F=50:T=0:S=0
30 A= RND 9:B= RND 9
40 L= ABS (X-A)+ ABS (Y-B)
50 IF X=A AND Y=B GOTO 400
100 IF L=1 BEEP 3
110 IF L=2 BEEP 2
120 IF L=3 BEEP 1
130 WAIT 40: PRINT "("; STR$(X);", "; STR$(Y);") A=";
  STR$(L);" E="; STR$(E)
150 S=S+1:E=F- INT (S/2)
153 IF E<=0 THEN 500
155 G$= INKEY$: IF G$=" " THEN 130
160 IF G$="2" THEN LET Y=Y-1: GOTO 210
170 IF G$="4" THEN LET X=X-1: GOTO 210
180 IF G$="6" THEN LET X=X+1: GOTO 210
190 IF G$="8" THEN LET Y=Y+1: GOTO 210
195 PAUSE "FALSCHE TASTE !"
200 GOTO 130
210 IF X<0 LET X=0: GOTO 130
220 IF Y<0 LET Y=0: GOTO 130
230 IF X>9 LET X=9: GOTO 130
240 IF Y>9 LET Y=9: GOTO 130
250 IF X=A AND Y=B GOTO 400
260 E=F- INT (S/2)
270 IF E<=0 GOTO 500
280 R= RND 5
290 IF R=1 LET B=B-1: GOTO 340
300 IF R=2 LET A=A-1: GOTO 340
310 IF R=3 LET A=A+1: GOTO 340

```

```

320 IF R=4 LET B=B+1: GOTO 340
340 IF A<0 OR A>9 GOTO 370
350 IF B<0 OR B>9 GOTO 370
360 GOTO 40
370 CALL &45E8: CALL &45E8
375 WAIT 6: FOR N=1 TO 4: PRINT ")))VIBRIER((((": PRINT
"(((VIBRIER)))": NEXT N: GOTO 30
400 PAUSE "<<< GEFANGEN >>>"
420 WAIT 0: PRINT "###>QUETSCH<###": CALL 1442: GOSUB 600
430 T=T+1: C= RND 3*5: F=F+C
435 E=F- INT (S/2)
440 WAIT 100: PRINT "PKT: "; STR$ (T); " * EN: "; STR$ (E)
450 GOTO 30
500 WAIT 0: PRINT "PUNKTE: "; STR$ (T): CALL 1442:
GOSUB 700: PRINT "-----ENDE-----": CALL 1442:
GOSUB 700: GOTO 5
600 FOR N=150 TO 0 STEP -10: POKE &45E9, N: CALL &45E8:
NEXT N: POKE &45E9, 255: RETURN
700 FOR N=50 TO 0 STEP -5: POKE &45E9, N: CALL &45E8:
NEXT N: POKE &45E9, 255: RETURN

```

Siebzehn und Vier

Beschreibung:

Dieses Programm ist eine Variante des bekannten Glücksspiels 17+4.

Ihr Gegenspieler ist dabei der Rechner. Es geht darum, daß man möglichst nahe mit der Summe seiner Karten an 21 herankommt. Erreicht man 21, hat man sofort gewonnen, ist man darüber, hat man sofort verloren (mit Ausnahme von 2 Assen (Feuer)). Möchte man z.B. mit 18 keine Karte mehr, beginnt der Rechner seine Karten zu nehmen. Um das Spiel interessanter zu gestalten, wurde der SHARP - PC so programmiert, daß er nur solange Karten nimmt, bis er mit seiner nächsten Karte mehr als 21 erreichen würde. Um diesen Vorteil des Rechners jedoch auszugleichen, gewinnt der Spieler bei Gleichstand.

Bekommt man mit den ersten 2 Karten 2 Asse, so ist dies "Feuer" - man hat gewonnen. Dasselbe gilt auch, wenn Sie 5 Bilder bekommen. Diese Regeln gelten sowohl für den Spieler, als auch für den Rechner.

Funktionsweise:

Sie starten das Programm mit «RUN». Es erscheint "**** 17 + 4 ****" "I CH MISCH E *". Nach dem Mischen bekommen Sie die ersten 2 Karten. Hinter der Karte, z.B. "KOENIG (14)", wird Ihnen Ihre bisherige Summe angezeigt - hier 14. Nach den ersten beiden Karten blinkt dann "*KARTE? «J/N»*". Drücken Sie «J», bekommen Sie nochmals eine Karte und es blinkt wieder "*KARTE? «J/N»*". Drücken Sie «S», wird Ihnen nochmals kurz Ihre Summe angezeigt. Drücken Sie «N», beginnt der Rechner seine Karten zu nehmen. Hat er sein Spiel beendet, wird angezeigt, wer gewonnen hat und danach, wer wieviele Spiele gewonnen hat, wobei "SP:" den Spieler und "PC:" den Rechner darstellt. Mit der Anzeige "I CH MISCH E *", beginnt ein neues Spiel.

<u>Beispiel:</u>	Rechner:	Eingabe des Spielers:
	"**** 17 + 4 ****"	
	"** ICH MISCHE **"	
	(BEEP)	
	"KOENIG (4)"	
	"NEUN (13)"	
	"*KARTE? «J/N»*"	«S»
	"*SUMME: (13)*"	
	"*KARTE? «J/N»*"	«J»
	"KOENIG (17)"	
	"*KARTE? «J/N»*"	«N»
	"SIEBEN (7)"	
	"ZEHN (17)"	
	"KARTEN REICHEN"	
	"**GLEICHSTAND**"	
	(BEEP)(BEEP)(BEEP)	
	"***SPIELER +1***"	
	(BEEP)	
	"SPIELER GEWINNT"	
	" SP:1 / PC:0 "	
	"** ICH MISCHE **"	
	usw.	

Listing:

```

10 DIM K$(32)*8:G=0:RR=0:SS=0: PAUSE " **** 17+4 ****"
15 PAUSE " **ICH MISCHE**": WAIT 15: RANDOM
20 FOR N=1 TO 32: READ K$(N): IF INT (N/8)=N/8
   THEN RESTORE
25 NEXT N
30 FOR N=1 TO 12
40 Y= RND 32:K$(0)=K$(N):K$(N)=K$(Y):K$(Y)=K$(0)
50 NEXT N: BEEP 1
100 S=0:G=0:B=0
110 FOR N=1 TO 6
120 IF N<3 THEN 150
130 PRINT " *KARTE? <J/N>*":J$= INKEY$: IF J$<>"J"
   AND J$<>"N"
   AND J$<>"S" THEN 130
135 IF J$="S" THEN PAUSE " *SUMME: (; STR$(S);)"*:
   GOTO 130
140 IF J$="N" THEN LET N=6: NEXT N: GOTO 300
150 S=S+ VAL ( RIGHT$( K$(N),2)): PAUSE " ";
   LEFT$( K$(N), LEN K$(N)-2);" (; STR$(S);)"
160 IF S=22 AND N=2 THEN LET N=6: NEXT N: PAUSE
   " ***FEUER***": LET G=1: GOTO "E"
170 IF S=21 THEN LET N=6: NEXT N: LET G=1: GOTO "E"
180 IF S>21 THEN LET N=6: NEXT N: LET G=2: GOTO "E"
185 IF VAL ( RIGHT$( K$(N),2))<5 THEN LET B=B+1
190 IF B=5 THEN LET N=6: NEXT N: PAUSE " ***5 BILDER***":
   LET G=1: GOTO "E"
200 NEXT N: END
300 R=0:B=0
310 FOR N=7 TO 12
320 R=R+ VAL ( RIGHT$( K$(N),2)): PAUSE " ";
   LEFT$( K$(N), LEN K$(N)-2);" (; STR$(R);)"
330 IF R=21 THEN LET N=12: NEXT N: LET G=2: GOTO "E"

```

```

340 IF R=22 AND N=8 THEN LET N=12: NEXT N: PAUSE
    " ***FEUER***": LET G=2: GOTO "E"
345 IF VAL ( RIGHTS ( K$(N), 2)) < 5 THEN LET B=B+1
350 IF B=5 THEN LET N=12: NEXT N: LET G=2: PAUSE
    " ***5 BILDER***": GOTO "E"
360 IF R+ VAL ( RIGHTS ( K$(N+1), 2)) > 21 AND N > 7
    THEN LET N=12
370 NEXT N
380 PAUSE " KARTEN REICHEN"
400 "E" IF G <> 0 THEN 440
410 IF S > R THEN LET G=1: GOTO 440
420 IF S=R THEN LET G=1: PAUSE " **GLEICHSTAND**": BEEP 3:
    PAUSE " ***SPIELER +1***": GOTO 440
430 G-2
440 IF G=1 THEN BEEP 1: PAUSE "SPIELER GEWINNT!": SS=SS+1:
    GOTO 460
450 IF G=2 THEN BEEP 2: PAUSE "PC-1401 GEWINNT!": RR=RR+1
460 PAUSE " SP: "; STR$(SS); " / PC: "; STR$(RR): GOTO 15
500 DATA "AS11", "ZEHN10", "NEUN09", "ACHT08", "SIEBEN07",
    "KOENIG04", "DAME03", "BUBE02"

```

Mastermind

Beschreibung:

Das Spiel ist als **MASTERMIND** oder **SUPERHIRN** bekannt. Ziel ist es die richtige vierstellige Zahlenkombination zu erraten.

Bei der Eingabe von **VARIATIONEN** kann man sich auf ein bestimmtes Zahlenintervall beschränken (z.B. werden nach Eingabe einer '6' die zu erratenden Ziffern im Bereich von 0...6 liegen).

Bei **WIEDERHOLUNG** wird als Eingabe 'J' oder 'N' erwartet; falls man 'J' eingibt, kann die zu erratende Zahlenkombination auch mehrere gleiche Ziffern beinhalten.

Bei Eingabe von 'N' ist keine Zahl in der Kombination doppelt vorhanden. Die Eingabe der Zahlenkombination erfolgt ohne Komma's oder Leerzeichen (z.B. '0123' oder '6577'), danach wird angezeigt wieviel Zahlen auf der richtigen Position sind und wieviele richtig, aber auf der falschen Position sind.

Benötigt man mehr als 11 Schritte, so wird das Spiel abgebrochen.

Listing:

```

1000 "Z"
1010 CLEAR:WAIT 300
1020 PRINT " ** MASTERMIND **"
1030 BEEP 1
1040 INPUT "VARIATIONEN? ";V
1045 IF V<1 OR V>9 GOTO 1040
1050 Z=0:RANDOM
1052 IF V<3 GOTO 1060
1055 INPUT "WIEDERHOLUNG? ";S$
1060 FOR I=1 TO 4:A(I)=RND(V+1)-1:NEXT I
1065 IF S$="N" GOSUB "CHECK"
1070 "INPUT"
1080 PAUSE "EINGABE:"
1090 INPUT "? ";Z$
1100 FOR I=1 TO 4

```

```
1110 A(I+4)=VAL(MIDS(Z$,I,1))
1120 NEXT I
1130 W=W+1
1140 FOR I=1 TO 4
1150 IF A(I+4)=A(I)LET X=X+1:A(I+10)=-1:A(I+14)=-1
1160 NEXT I
1170 FOR I=1 TO 4
1180 FOR J=1 TO 4
1185 IF A(10+J)<>0 OR A(14+I)<>0 GOTO 1200
1190 IF A(I+4)=A(J)LET Y=Y+1:A(J+10)=-1:A(I+14)=-1:GOTO 1205
1200 NEXT J
1205 NEXT I:WAIT
1210 PRINT Z$;" ";X;" ";Y
1220 IF X=4 GOTO "WIN"
1230 IF W>10 GOTO "LOOSE"
1235 FOR U=5 TO 18:A(U)=0:NEXT U
1240 X=0:Y=0
1250 GOTO "INPUT"
1260 "WIN"
1270 PRINT W;" SCHRITTE"
1280 PAUSE "NOCH EIN ..."
1290 INPUT "SPIEL (J/N)? ";US
1300 IF US="J" GOTO "Z"
1310 END
1320 "CHECK"
1325 FOR I=1 TO 4
1330 A(I)=RND(V+1)-1
1340 FOR T=1 TO I-1
1345 IF I=1 GOTO 1360
1350 IF A(I)=A(T)GOTO 1330
1360 NEXT T
1365 NEXT I
1370 RETURN
1380 "LOOSE"
1385 PAUSE "11 SCHRITTE":Z$=""
1390 FOR I=1 TO 4
1400 Z$=Z$+STR$(A(I)):NEXT I
1410 PRINT Z$
1420 GOTO 1280
```

Denktest

Beschreibung:

Dieses Programm ist zur Abwechslung auch für verschiedene Sharp Rechner gedacht. Nur ein 24 Zeichen-Display wäre als Bedingung für die Lauffähigkeit dieses Programmes zu sehen.

Während des Ablaufs des Programmes werden 3 Zeichenfolgen eine kurze Zeitlang gleichzeitig dargestellt. Es gilt nun klaren Kopf zu behalten und möglichst viel der Kombinationen zu merken. Gleich darauf wird man aufgefordert die drei Zeichenfolgen hintereinander einzugeben. Je nach Merkfähigkeit wird man vom Computer in seiner Intelligenz eingestuft.

Das Programm war ursprünglich für den Sharp 1251 gedacht. Mit ein paar PRINTs und WAITs ist es nun jedoch auf jeden Fall auf dem Sharp PC-1350 lauffähig. Wenn man die Anzahl der Kombinationen von 3 auf 2 kürzt, kann man das Spiel auch auf einem Sharp der 14er Serie (14XX) laufen lassen.

Listing:

```

1000 "G":USING:CLS:CLEAR:RANDOM
1010 WAIT 100:PRINT "*** Gedaechnistest ***"
1020 PRINT "Merke die Kombinationen"
1030 PRINT "und gebe sie dann ein!"
1040 PRINT "----- Viel Glueck -----"
1050 DIM G$(6)*1,N$(10)*1,V$(3)*3,X$(3)*6,Z$(3)*3,
    Y$(3)*6
1060 FOR I=1 TO 9:N$(I)=STR$ I:NEXT I:N$(10)="0"
1070 FOR I=1 TO 6
1080 J=RND 26:J=J+64
1090 G$(I)=CHR$(J):NEXT I
1100 FOR I=1 TO 3
1110 Y$(I)=" "
1120 FOR J=1 TO 3
1130 K=RND 9
1140 Y$(I)=Y$(I)+N$(K):NEXT J
1150 L=RND 9:J=(I-1)*2+1
1160 A$(I)=G$(J)+G$(J+1)+N$(L)
1170 H$=Y$(I):A$(I+3)=RIGHT$(H$,3):NEXT I
1180 CLS:WAIT 50
1190 PRINT "Achtung!":WAIT 0:PRINT " "
1200 GOSUB 1600
1210 CLS
1220 FOR I=1 TO 3
1230 ON I GOTO 1240,1250,1260
1240 INPUT "1.Antwort ? ";X$(I):GOTO 1270
1250 INPUT "2.Antwort ? ";X$(I):GOTO 1270
1260 INPUT "3.Antwort ? ";X$(I)
1270 X$(I)=LEFT$(X$(I),6)
1280 Z$(I)=LEFT$(X$(I),3)
1290 V$(I)=RIGHT$(X$(I),3)
1300 NEXT I
1310 CLS:WAIT 50
1320 PRINT "Aufgabe:"
1330 GOSUB 1600
1340 WAIT 50
1350 PRINT "Dein Ergebnis:"
1360 GOSUB 1700
1370 N=1
1380 FOR I=1 TO 3
1390 IF A$(I)=Z$(I) LET N=N+1
1400 IF A$(I+3)=V$(I) LET N=N+1
1410 NEXT I
1420 CLS:WAIT 0:PRINT " "
1430 WAIT 150:ON N GOTO 1440,1450,1460,1470,1480,
    1490,1500
1440 BEEP 1:PRINT " Du bist ein Vollidiot":GOTO 1510
1450 BEEP 1:PRINT " Das war ganz schlecht":GOTO 1510
1460 BEEP 2:PRINT " Maessige Leistung ":GOTO 1510
1470 BEEP 2:PRINT " Immerhin, ganz gut ":GOTO 1510
1480 BEEP 3:PRINT " Na also, sehr gut ":GOTO 1510
1490 BEEP 4:PRINT "** Kluges Koepfchen **":GOTO 1510
1500 BEEP 5:PRINT " ***** Genie *****"
1510 WAIT 0:PRINT " "
1520 W$="":BEEP 1:INPUT " Nochmal (J/N) ? ";W$

```

```

1530 IF W$="N" THEN 1740
1540 IF W$="J" THEN 1070
1550 GOTO 1510
1600 WAIT 200:BEEP 2:PRINT A$(1);A$(4);"  ";A$(2);A$(5);
"  ";A$(3);A$(6)
1610 RETURN
1700 WAIT 0:BEEP 1:PRINT USING "&&&&&";X$(1);
USING;"  ";
1710 WAIT 200:PRINT USING "&&&&&";X$(2);USING;"  ";
USING "&&&&&";X$(3)
1720 USING
1730 RETURN
1740 CLS:END

```

51 Gleich Tot

Beschreibung:

Dieses Spiel simuliert ein ganz normales Skatblatt mit 32 Karten. Von einem angeordneten Block (die 32 Karten werden übereinanderliegend dargestellt) werden abwechselnd für den Spieler und den Computer jeweils 4 Karten abgehoben. Die dem Spieler gehörenden Karten werden links vom Kartenblock aufgedeckt. Der Computer behält seine Karten natürlich für sich.

Der Spieler beginnt nun mit einer der Tasten W,E,R oder T die sich exakt darüber befindende Karte auszuwählen, die dann automatisch verschwindet und rechts auf dem "Zählblock" erscheint. Der Computer zählt den Wert der Karte, und setzt an die Stelle der verschwundenen Karte eine Neue, die vom Kartenblock abgehoben wurde.

Nun ist der Computer selbst am Zuge. Aus seinen 4 Karten wählt er eine aus und legt sie dann sichtbar für den Spieler auf den Zählblock. Rechts am Zählblock wird der Wert der Karte zum bisherigen Punktestand hinzugezählt.

Verloren hat derjenige Spielpartner (Computer oder Spieler), der eine Karte auf den Zählblock legen muß, mit der der Punktestand über 50 ansteigt. (Ganz entsprechend dem Titel des Spiels 51=TOT)

Der Gag an dem Spiel entsteht durch die verschiedenen Punkte, die die Karten wert sind. Die meisten Karten haben Punktwerte wie im Skatspiel. Ausnahmen bilden nur die "9", die "10" und das "As": Die "9" hat den Wert null. Somit ändert sich am Punktestand überhaupt nichts, wenn diese Karte auf den Zählblock gelegt wird. Die "10" hat den Wert -10. Auf diese Weise ist es möglich, den Punktestand, sofern er bedrohlich nah an die 50 herankommt, auch zu reduzieren.

Das "As" hat wahlweise den Wert 11 oder 1. Jedoch nur wenn beides möglich ist, ohne daß der Punktestand über 50 ansteigt, wird dem Spieler die Wahl gelassen. An der Stelle des As erscheint eine 11, die man annimmt, indem man die Taste "J" für Ja drückt. Bei "N" für Nein wird dem Punktestand nur 1 hinzugezählt.

Listing:

```

100 "D":CLEAR:CLS:WAIT 0:RESTORE
110 CURSOR 32:PRINT "51 = Tot"
120 LINE(30,4)-(120,18),B
130 CURSOR 73:PRINT "Karten werden gemischt"
140 DIM W$(3)*14,K$(8)*14,T$(32)*2,S$(3)*2,A$(3)*2,A(3)
150 FOR I=0 TO 3:READ W$(I):NEXT I
160 FOR I=0 TO 8:READ K$(I):NEXT I
170 FOR I=0 TO 7
180 T$(I+1)="0"+STR$ I
190 T$(I+9)="1"+STR$ I
200 T$(I+17)="2"+STR$ I
210 T$(I+25)="3"+STR$ I
220 NEXT I
230 USING "###":C=32:ZZ=0:R=0:CLS:BEEP 2:
    PAUSE "Spielbeginn":CLS
240 LINE(97,31)-(85,0),X,BF
250 FOR I=0 TO 3:GOSUB 480:GOSUB 340:GOSUB 480:GOSUB 560
260 NEXT I
270 LINE(110,9)-(122,31),B:BEEP 1
280 Z$="":Z$=INKEY$
290 IF Z$="W" LET I=0:GOTO 380
300 IF Z$="E" LET I=1:GOTO 380
310 IF Z$="R" LET I=2:GOTO 380
320 IF Z$="T" LET I=3:GOTO 380
330 GOTO 280
340 LINE(11+I*17,9)-(23+I*17,31),B
350 GCURSOR(14+I*17,19):GPRINT W$(VAL LEFT$(D$,1))
360 GCURSOR(14+I*17,28):GPRINT K$(VAL RIGHT$(D$,1)):
    S$(I)=D$
370 RETURN
380 LINE(11+I*17,9)-(23+I*17,31),R,BF
390 GCURSOR(113,19):GPRINT W$(VAL LEFT$(S$(I),1))
400 GCURSOR(113,28):GPRINT K$(VAL RIGHT$(S$(I),1))
410 PP=VAL RIGHT$(S$(I),1):S$(I)=" "
420 IF PP>4 LET PP=PP+2:IF PP=9 LET PP=-10
430 IF PP=1 AND ZZ<40 GCURSOR(14+I*17,24):GPRINT K$(8):
    GOSUB 600
440 GOSUB 640
450 IF ZZ>50 BEEP 2:CLS:PAUSE " V E R L O R E N !":
    GOTO 510
460 IF C GOSUB 480:GOSUB 340
470 GOTO 650
480 RANDOM:Y=RND C:D$=T$(Y):T$(Y)=T$(C):T$(C)=D$:C=C-1
490 LINE(85,R)-(97,R),R,BF:R=R+1
500 RETURN
510 PRINT " ":PRINT " Neues Spiel ? (J/N)"
520 Z$="":Z$=INKEY$:IF Z$<>"J" AND Z$<>"N" THEN 520
530 IF Z$="J" GOTO 230
540 CLS:USING:WAIT:END
550 REM ENDE
560 A$(I)=D$:A(I)=VAL RIGHT$(D$,1)
570 IF A(I)>4 LET A(I)=A(I)+2:IF A(I)=9 LET A(I)=-10
580 IF A(I)=1 AND ZZ<40 LET P1=RND 2:IF P1=2 LET A(I)=11
590 RETURN
600 Z$="":Z$=INKEY$

```

Barrikade

Beschreibung:

Der Sinn des Spieles ist natürlich klar. Mit dem Paddle soll der Ball so lange wie möglich im Spielfeld gehalten werden, wobei es beim Abräumen der vorderen Pointfelder selbstverständlich Punkte gibt.

Die Steuerung ist auch recht einfach. Man stelle sich den Zahlenblock des PC-1350 als Steuerpult vor. Die um die "5" herum angeordneten Zahlen deuten in die Richtung in die der Paddle gesteuert werden soll. (Beispiel: Die "2" steuert den Paddle direkt nach unten, die "9" nach rechts oben.) Extrafunktion besitzt die Taste "5" selbst. Wenn sie betätigt wird springt der Paddle von jeder Position aus an eine vorgegebene Position, die sich etwa in der Mitte des Bewegungsfeldes befindet.

Listing:

```

2000 "B":CLS:CLEAR:USING "###"
2005 WAIT 0:CURSOR 31:PRINT "Barrikade"
2006 WAIT 150:LINE(44,4)-(104,18),B
2007 CLS:WAIT 0:CURSOR 18:PRINT "Punkte"
2010 LINE(100,0)-(0,0):LINE(0,1)-(0,31):LINE-(100,31)
2015 LINE(5,1)-(40,30),BF:B=0
2016 E=RND(34)+5:R=RND(34)+5:N=RND(30):M=RND(30):B=B+1
2017 LINE(E,N)-(R,M),X,BF:IF B=12 GOTO 2020
2018 GOTO 2016
2020 X=RND(5)+40:Y=RND(29)+1
2025 A=50:B=14:D=2:T=2100:K=140:L=0
2030 D$=INKEY$:D=VAL D$
2035 IF D=0 GOTO 2060
2040 D=2040+D:GOTO D
2041 A=A-1:B=B+1:GOTO 2050
2042 B=B+1:GOTO 2050
2043 A=A+1:B=B+1:GOTO 2050
2044 A=A-1:GOTO 2050
2045 A=80:B=14:GOTO 2050
2046 A=A+1:GOTO 2050
2047 A=A-1:B=B-1:GOTO 2050
2048 B=B-1:GOTO 2050
2049 A=A+1:B=B-1
2050 IF A<42 LET A=42:GOTO 2070
2051 IF A>100 LET A=100
2054 IF B<1 OR B>28 LET B=U
2055 LINE(O,U)-(O,U+2),R
2060 LINE(A,B)-(A,B+2):O=A:U=B
2070 PRESET(K,L)
2080 PSET(X,Y):K=X:L=Y:GOTO T
2100 X=X+1:Y=Y+1:H=2110:J=2120:I=2130:GOTO 2200
2110 X=X-1:Y=Y+1:H=2100:J=2130:I=2120:GOTO 2200
2120 X=X+1:Y=Y-1:H=2130:J=2100:I=2110:GOTO 2200
2130 X=X-1:Y=Y-1:H=2120:J=2110:I=2100:GOTO 2200
2140 X=X+1:H=2150:GOTO 2200
2150 X=X-1:H=2140:GOTO 2200
2200 IF POINT(X,L)LET T=H:GOTO 2250
2205 IF T>2130 GOTO 2230
2210 IF POINT(K,Y)LET T=J:GOTO 2300

```

```

2220 IF POINT(X,Y)LET T=I:GOTO 2320
2230 GOTO 2030
2250 IF X>K AND X=0 GOTO 2350
2260 IF X<5 OR X>40 GOTO 2340
2270 IF L>0 AND L<31 PRESET(X,L):C=C+1:BEEP 1:GOTO 2340
2300 IF K=0 GOTO 2400
2305 IF K<5 OR K>40 GOTO 2340
2310 IF Y>0 AND Y<31 PRESET(K,Y):C=C+1:BEEP 1:GOTO 2340
2320 IF X=0 AND X>K GOTO 2350
2325 IF X<5 AND X>40 GOTO 2340
2330 IF Y>0 AND Y<31 PRESET(X,Y):C=C+1:BEEP 1:GOTO 2340
2340 CURSOR 92:PRINT C:X=K:Y=L:GOTO T
2350 IF L=B+1 LET T=2150:GOTO 2380
2360 IF L=B OR Y=B LET T=2130:GOTO 2380
2370 T=2110
2380 Y=L:X=K:GOTO T
2400 IF Y>L LET T=2120:GOTO 2430
2410 IF Y<L LET T=2100:GOTO 2430
2420 GOTO 2030
2430 X=K:Y=L:GOTO T
2440 END
610 IF (Z$<>"J") AND (Z$<>"N") GOTO 600
620 IF Z$="J" LET PP=11
630 LINE(14+I*17,16)-(21+I*17,24),R,BF:RETURN
640 ZZ=ZZ+PP:CURSOR 93:PRINT ZZ:RETURN
650 FOR J=0 TO 3:FOR K=0 TO 2
660 IF A(J)>A(K) THEN 680
670 GOTO 700
680 U=A(J):A(J)=A(K):A(K)=U
690 V$=A$(J):A$(J)=A$(K):A$(K)=V$
700 NEXT K
710 NEXT J
720 FOR I=0 TO 3:IF I<3 GOTO 740
730 GOTO 750
740 IF A(I)=0 AND A(I+1)=-10 LET P1=RND 2:
IF P1=1 GOTO 820
750 IF (A(I)=11) AND (ZZ>39) LET A(I)=1:I=I+12:GOTO 820
760 IF A(I)+ZZ<51 GOTO 780
770 GOTO 820
780 GCURSOR(113,19):GPRINT W$(VAL LEFT$(A$(I),1))
790 GCURSOR(113,28):GPRINT K$(VAL RIGHT$(A$(I),1)):
A$(I)=""
800 PP=A(I):GOSUB 640:A(I)=50
810 I=I+5
820 NEXT I
830 IF I>10 GOTO 650
840 IF I=4 BEEP 3:CLS:PAUSE " GEWONNEN !":
GOTO 510
850 I=I-1:IF I>3 LET I=I-5
860 IF C GOSUB 480:GOSUB 560
870 K=0:FOR I=1 TO 4
880 PP=VAL RIGHT$(S$(I-1),1):IF PP>4 LET PP=PP+2:
IF PP=9 LET PP=-10
890 IF (S$(I-1)="") OR (ZZ+PP>50) LET K=K+I
900 NEXT I
910 IF K=10 LET ZZ=51:GOTO 430
920 BEEP 1:GOTO 280
930 DATA "081C4A774A1C08","185C7E7F7E5C18"
940 DATA "0E1F3F7E3F1F0E","081C3E7F3E1C08"

```

```

950 DATA "00064949291E00", "007C1211127C00"
960 DATA "00417F49493600", "00417F41413E00"
970 DATA "007F0814224100", "00010179050300"
980 DATA "00364949493600", "7F003E4141413E"
990 DATA "427F4000427F40"

```

Mondlandung

Beschreibung:

Mondlandung ist zwar auch schon wieder ein altes Thema, jedoch immer wieder sehr reizvoll zu spielen.

Mondlandung wird mit Def "M" oder RUN gestartet. Um das Spiel zu beginnen drücke man eine beliebige Taste, die man dann auch zum "Gasgeben" verwendet. Ziel des Spiels ist es natürlich, das kleine Raumschiffchen möglichst behutsam auf dem Mond aufzusetzen. Nach einer Superlandung oder einem Crash kann man das Spiel mit der "M"-Taste neu beginnen.

Einen Tip möchte ich diesem Spiel mitgeben. Es dürfte von Vorteil sein, wenn Ihr Euren PC-1350 bevor Ihr loslegt an das Netz anschließt, da "Mondlandung" relativ viel Batterie benötigt. Für ein oder zwei Mal ist das wahrscheinlich nicht notwendig, jedoch wird es wahrscheinlich jeder solange versuchen bis er das Schiffchen zumindest einmal sanft gelandet hat.

Listing:

```

100 "M":WAIT 0:CLEAR:CLS
110 DIM Z$(6):DIM F$(3)
120 REM MOND-LANDUNG
130 GCURSOR(6,31):GPRINT "22362A2A222222"
140 GCURSOR(6,23):GPRINT "0000728B8A8A72"
150 GCURSOR(6,15):GPRINT "0000C629282827"
160 GCURSOR(6,7):GPRINT "808080BC808080"
170 GCURSOR(16,31):GPRINT "80809DA5A5A59D"
180 GCURSOR(16,23):GPRINT "010167A9292927"
190 GCURSOR(18,15):GPRINT "4A4B4A5A6A0000"
200 GCURSOR(18,7):GPRINT "CE5252524E021C"
210 Z$(1)="183C5E7EDB8181"
220 Z$(2)="183C6E7EDB8181"
230 Z$(3)="183C767EDB8181"
240 Z$(4)="183C7A7EDB8181"
250 Z$(5)="183C7C7EDB8181"
260 Z$(6)="183C3E7EDB8181"
270 F$(1)="9981"
280 F$(2)="99A5"
290 F$(3)="81A5"
300 S=0:G=0:V=0:H=27:M=27:W=0:U=310
310 IF INKEY$ LET W=0.2:U=320:GOSUB 600
320 S=S+1:V=V+W:H=H+V:P=INT H
330 IF M=P GOTO 350
340 GCURSOR(M,19):GPRINT "0000000000000000"
350 GCURSOR(P,19):GPRINT Z$(S)
360 IF INKEY$ LET G=G+1:GCURSOR(P+5,19):GPRINT F$(G):
V=V-0.3
370 IF G>2 LET G=0

```

```
380 IF S>5 LET S=0
390 M=P
400 IF P<140 AND P>-7 GOTO U
410 IF P>139 AND V<0 GOTO U
420 IF P>139 AND V>0.3 GOSUB 700:GOTO 450
430 IF P<-6 GOSUB 800:GOTO 450
440 GOSUB 900:REM SUPERLANDUNG
450 T$=INKEY$:IF T$="" GOTO 450
460 IF T$="M" GOTO 100
470 IF T$="E" GOTO 490
480 GOTO 450
490 CLS
500 END
600 CLS:REM LANDEPLATZ
610 GCURSOR(P,19):GPRINT Z$(S)
620 FOR K=1 TO 3
630 LINE(150-K,31)-(150-K,0)
640 NEXT K
650 RETURN
700 REM CRASH
710 GCURSOR(139,31):GPRINT "3844414141453900"
720 GCURSOR(139,23):GPRINT "0000678808080700"
730 GCURSOR(139,15):GPRINT "00008F908E815E00"
740 GCURSOR(139,7):GPRINT "4040784444444400"
750 RETURN
800 REM VERSCHOLLEN
805 GCURSOR(6,31):GPRINT "808081828282F900"
810 GCURSOR(6,23):GPRINT "0000C7282720CF00"
815 GCURSOR(6,15):GPRINT "0010B81010940800"
820 GCURSOR(6,7):GPRINT "4000565951515100"
825 GCURSOR(16,31):GPRINT "1C22201C02221C00"
830 GCURSOR(18,23):GPRINT "F18A8A8AF1808000"
835 GCURSOR(18,15):GPRINT "E3242424F3000000"
840 GCURSOR(18,7):GPRINT "9C223E209C000000"
845 FOR M=1 TO 3
850 FOR N=3 TO 0 STEP-1
855 PSET(N,36-M*10-N):PSET(N,36-M*10+N)
860 NEXT N:NEXT M
865 RETURN
900 REM SUPERLANDUNG
910 GCURSOR(6,31):GPRINT "708882720A8A7100"
920 GCURSOR(8,23):GPRINT "2F282868AF080800"
930 GCURSOR(8,15):GPRINT "1CA2BEA01C000000"
940 GCURSOR(8,7):GPRINT "B0C08F8080000000"
950 GCURSOR(16,31):GPRINT "80809DA5A5A59D"
960 GCURSOR(16,23):GPRINT "010167A9292927"
970 GCURSOR(18,15):GPRINT "4A4B4A5A6A0000"
980 GCURSOR(18,7):GPRINT "CE5252524E021C"
990 RETURN
```

Hangman

Beschreibung:

Auf besonderen Wunsch des Autors werden hier an dieser Stelle Adresse und Anschrift genannt:

Hendrik Ambrosius Cube-Soft Goerdelerstr. 29 2400 Lübeck 1

Dies ist eine weitere Variante des bekannten Hangman-Spiels, bei dem es darum geht, mit möglichst wenig Versuchen ein Wort zu erraten. Der Reiz bei dieser Version liegt darin, daß man auch mit seinem Freund um die Wette spielen kann, indem man sich gegenseitig Wörter vorgibt. Man kann aber natürlich auch allein spielen und der Computer sucht per Zufall ein Wort aus der Liste aus.

Die Bedienung ist sehr einfach und dürfte sich wohl selbst erklären. Nur noch ein paar Anmerkungen: wenn "(*=ENDE)" erscheint, wartet der Rechner auf die Eingabe eines Buchstabens oder des ganzen Wortes. Die Eingabe wird dann bearbeitet.

Das erste Mal muß das Programm mit "RUN" gestartet werden, alle weiteren Spiele kann man den Vorspann mit "A" überspringen.

Listing:

```

10 DIM A$(0)*14,B$(0)*14,C$(0)*14,D$(0)*14,H$(0)*80:
   Q=25
15 " "
20 RESTORE 800
30 READ H$(0)
40 FOR I=1 TO LEN(H$(0)):WAIT 20:PRINT MID$(H$(0),I,16):
   NEXT I:WAIT 100 "A"
102 PRINT "RATEWORT...":PRINT "1)SELBST EINGEB.":
   PRINT "2)ZUFALLSWAHL":INPUT "1 ODER 2:":W
104 IF W<>1 THEN 107
105 INPUT "WORT:":A$(0):IF LEN(A$(0))<2 OR LEN(A$(0))>14
   THEN GOTO 105
106 GOTO 110
107 RANDOM:Y=RND Q:RESTORE 610:FOR I=1 TO Y:READ A$(0):
   NEXT I
110 L=LEN(A$(0)):B$(0)=" "
120 Z=0:V=0:FOR I=1 TO L
130 B$(0)=B$(0)+"-"
140 NEXT I
200 "S"
202 IF V=L*1.5+2 THEN "J"
205 V=V+1
210 WAIT 0:PRINT " ";B$(0):CALL &141F:WAIT:
   INPUT "(*=ENDE):":C$(0)
220 IF C$(0)="*" THEN "F"
230 K=LEN(C$(0))
240 IF K<1 OR K>L THEN BEEP 1:PRINT "WAS SOLL DAS ??":
   GOTO "S"
250 IF K=1 THEN "D"

```

```

260 IF C$(0)=A$(0)THEN BEEP 3:CALL 1444:BEEP 2:
    PRINT "BRAVO!! RICHTIG.":PRINT "VERSUCHE: ";V:
    SC=INT(2*L/V)
265 IF C$(0)=A$(0)THEN PRINT "PUNKTE: ";SC:GOTO "A"
270 PRINT "DAS IST LEIDER":PRINT "NICHT KORREKT!":
    GOTO "S"
280 "D"
290 FOR I=1 TO L
300 IF C$(0)=MID$(B$(0),I,1)THEN PRINT "WURDE SCHON":
    PRINT "EINMAL GEFRAGT!":GOTO "S"
310 NEXT I
320 D$(0)="":X=0
330 FOR I=1 TO L
335 O$(0)=MID$(A$(0),I,1)
340 IF C$(0)=O$(0)THEN LET D$(0)=D$(0)+O$:Z=Z+1:X=1
350 IF C$(0)<>O$(0)THEN LET D$(0)=D$(0)+MID$(B$(0),I,1)
360 NEXT I:B$(0)=D$(0)
370 IF X=0 THEN PRINT "LEIDER FALSCH!":GOTO "S"
380 PRINT "KOMMT VOR !"
385 IF Z<>L THEN "S"
386 BEEP 3:CALL 1444:BEEP 2:SC=INT(2*L/V)
390 PRINT "DAS RAETSEL IST":PRINT "GELOEST. DAS GE-":
    PRINT "SUCHTE WORT WAR:"
400 PRINT " ";A$(0)
410 PRINT "BRAVO! BRAVO!":PRINT "VERSUCHE: ";V:
    PRINT "PUNKTE: ";SC
420 GOTO "A"
430 "F"
440 PRINT "DU HAST NACH":PRINT V;"VERSUCHEN":
    PRINT "AUFGEGEBEN.DAS"
450 PRINT "RICHTIGE WORT":PRINT "WAERE GEWESEN:":
    PRINT " ";A$(0)
460 PRINT "ICH WUENSCHTE DIR":PRINT "MEHR GLUECK":
    PRINT "FUERS NAECHSTE":PRINT "MAL...":GOTO "A"
470 "J"
480 PRINT "NU IS ABER GENUG":PRINT V;"VERSUCHE":
    PRINT "HAETTEN AUSREI-"
490 PRINT "CHEN MUESSEN.DAS":PRINT "ZU FINDENDE WORT":
    PRINT "WAR GEWESEN:":PRINT " ";A$(0)
500 GOTO 460
610 DATA "TELEFON","COMPUTER","FERNSEHER",
    "RADIOSTATION","LANDKARTE"
620 DATA "FENSTER","OHRENSESSEL","SCHEINWERFER",
    "SALZSTANGEN","STAUBSAUGER"
630 DATA "SCHOKOLADE","ANKERKETTE","BUECHEREI",
    "KOFFERTRAEGER","WASCHMITTEL"
640 DATA "SCHIRMMUETZE","HUNDEHUETTE","KERZENLICHT",
    "BLUMENVASE","ZITRONENSAFT"
650 DATA "STRANDKORB","MANTELTASCHE","EISVERKAEUFER",
    "UHRENARM BAND","ELEFANTENOHR"
700 "M"
710 PRINT "DEF-BELEGUNGEN:":PRINT "A-SPIELSTART":
    PRINT ",-VORSPANN":PRINT "S-ABGEBROCHENES"
720 PRINT "SPIEL FORTSETZEN"
800 DATA "          HANGMAN FOR SHARP 14XX (C) 1987
    CUBE-SOFT+H.AMBROSIUS          "

```

Senso

Beschreibung:

Auf besonderen Wunsch des Autors werden hier an dieser Stelle Adresse und Anschrift genannt:

Hendrik Ambrosius Cube-Soft Goerdelerstr. 29 2400 Lübeck 1

Dieses Spiel ist sehr gut zum Gedächtnistraining geeignet. Man soll sich eine Zahlenreihe merken und darf dabei keinen Fehler machen.

Zur Bedienung:

Zuerst muß man die Geschwindigkeit eingeben. Anschließend leuchten dann Zahlen von 1 bis 4 auf. Nach jedem richtigen Versuch wird die Kette um eine Zahl länger. Man muß dann die ganze Zahlenreihe jedes Mal durch das Drücken der Tasten 1 bis 4 wieder eingeben. Wenn man einen Fehler macht, ist das Spiel zu Ende.

Listing:

```

10 DIM N(200),F$(0)*16
20 RESTORE:WAIT 10
30 READ F$(0):IF F$(0)="EOD" GOTO "A"
40 FOR I=1 TO LEN(F$(0))
50 PRINT LEFT$(F$(0),I)
60 NEXT I:PAUSE F$(0)
70 GOTO 30
80 DATA "* COMPISENSO 2 *","SHARP - VERSION"
90 DATA "(C) 86 CUBE-SOFT","AND H.AMBROSIUS","EOD"
100 "A"V=0
110 INPUT "SPEED(1-10):";S
120 IF S<1 OR S>10 GOTO 110
130 "R"V=V+1:RANDOM
140 N(V)=RND 4
150 FOR I=1 TO V:WAIT(11-S)*8
170 ON N(I)GOTO 180,190,200,210
180 PRINT " 1 * * *":GOTO 220
190 PRINT " * 2 * *":GOTO 220
200 PRINT " * * 3 *":GOTO 220
210 PRINT " * * * 4"
220 NEXT I
240 CALL 1444
250 PRINT " * * * *"
255 FOR I=1 TO V
260 IF INKEY$="" GOTO 260
270 A$=INKEY$
280 IF A$="1" THEN PRINT " 1 * * *"
290 IF A$="2" THEN PRINT " * 2 * *"
300 IF A$="3" THEN PRINT " * * 3 *"
310 IF A$="4" THEN PRINT " * * * 4"
320 IF VAL(A$)<>N(I)GOTO 330
325 NEXT I
327 CALL 1444:GOTO "R"
330 BEEP 2:WAIT 150:PRINT "FALSCH !"
340 PRINT "DU HAST BEIM":PRINT V;"VERSUCH"
350 PRINT "EINEN FEHLER":PRINT "GEMACHT."
360 INPUT "NOCHMAL(J/N)?";A$
370 IF A$<>"N" GOTO "A"

```


Shipdown

Beschreibung:

Wem das Spiel noch aus seiner Schulzeit bekannt ist, darf den folgenden Absatz ruhig überspringen.

Zu den Regeln:

Jeder Spieler hat 8 Schiffe, die über ein Spielfeld von 10*10 Karos beliebig verteilt sind. Darunter ist ein Vierer (ein Schiff aus vier Karos), zwei Dreier (aus drei Karos), drei Zweier und zwei Einer. Dabei dürfen sich keines der Schiffe überlappen oder um die Ecke liegen.

Jeder 'schießt' abwechselnd, indem er dem Gegner die Koordinaten des Zieles mitteilt, worauf dieser zurückgibt, ob an dem angegebenen Punkt ein Schiff liegt ('TREFFER') oder nur Wasser schwappt. Sind alle 'Bausteine' eines Schiffes getroffen, gilt das Schiff als 'versenkt'.

Je nachdem, wer zuerst dem Gegner alle Schiffe versenkt hat, hat gewonnen. Hat man getroffen, darf man noch einmal schießen, anderenfalls ist der Gegner an der Reihe. Der größte Aufwand bei diesem Spiel ist wohl das Zeichnen und Aufstellen der eigenen Schiffe auf einer karierten Seite.

Aufbau und Ablauf:

Der Rechner baut sich per Zufallsgenerator ein eigenes Feld auf, das jedoch direkt nach dem Einschalten immer gleich ist. Es empfiehlt sich daher, immer vor dem Start eine andere Zufallszahl generieren zu lassen.

Die Bezeichnung der Felder ergibt sich aus der Zeilennummer, gefolgt von der Spaltennummer. So ist z.B. '34' das Karo drei Zeilen nach unten und 4 Spalten nach rechts. Die Nummern gehen dabei von 0-9 und dürfen bei der Eingabe nicht getrennt werden.

Falls doch jemand mehrere Werte eingibt: der Rechner nimmt nur die beiden linken Zeichen der Eingabe und richtet sich nach deren Wert. So ergibt 'AB' die Werte '00', '1,2' ergibt 10, '12'=00 usw... .

Gestartet wird das Spiel mit 'RUN' oder Def 'A'. Dann wird das Feld aufgebaut und die Variablen eingerichtet. Ein Warmstart ist über Def 'B' oder falls mitten im Spiel abgebrochen wurde, mit GOTO 'START' möglich.

Der Spielablauf :

Aus reiner Höflichkeit läßt einem der Rechner immer den ersten Schuß. Es erscheint dann die Abfrage 'KOORDINATEN:', worauf man dann die beiden Werte seines Zieles eingibt (nicht getrennt ! s.o.).

Hat man getroffen, gibt's die entsprechende Treffermeldung und man darf noch einmal schießen.

Kommt dagegen die Nachricht 'WASSER !', hat man nichts erwischt und der Rechner ist an der Reihe.

'Schießt' der Rechner, so gibt er 'ZIEL : ' a,b aus, wobei 'a' die Zeilennummer und 'b' die Spalte des anvisierten Zieles ist.

Danach fragt der PC nach dem Ergebnis, worauf man entweder 'TREFFER', 'VERSENKT' oder, im Falle eines Fehlschusses, 'WASSER' eingeben sollte. Zur Erleichterung: der Rechner nimmt nur den Anfangsbuchstaben, wobei für ihn nur 'T' und 'V' relevant sind. Alles andere, auch einfach nur 'ENTER', zählt als 'Wasser'.

Natürlich schießt der Rechner bei einem Treffer noch einmal und läßt nicht eher von einem Ziel ab, bis er die Meldung 'Versenkt' erhalten hat. Für Mogler: Wird direkt nach dem Aufbau Def 'M' gedrückt, druckt der SHARP seinen Aufbau auf einen (wenn) angeschlossenen Thermodrucker (o.ä.) aus. Nichtbesitzer eines Druckers müssen dann in Zeile 790 aus dem 'LPRINT' ein 'PRINT' machen. Weiter geht's dann mit Def 'B'. Natürlich läßt sich dieses auch als Kontrolle nach dem Spiel benutzen.

Es ist übrigens ratsam, die eigenen Schüsse und die des Rechners mit aufzuzeichnen.

Kompatibilität:

Mit einem Speicherbedarf von 3431 Bytes ist das Programm auch auf dem PC 1401 lauffähig. Auch habe ich in dieser Version alle PEEK's und POKE's entfernt, so daß das Programm durchaus auch auf anderen Sharp-Rechnern laufen dürfte.

Das Programm funktioniert auch in Omikron-Basic, jedoch müssen vorher alle Marken, die in Anführungsstrichen stehen (außer bei den GOTO/GOSUB-Befehlen), mit einem Minuszeichen vorher und einem Doppelpunkt danach versehen werden. Auch sind die Anführungsstriche selbst zu löschen, (auch beim GOTO/GOSUB). Des weiteren sind nur noch alle BEEP X durch PRINT CHR\$(7) zu ersetzen.

Sontiges:

Ein paar kleine Fehlertest sind mit eingebaut, so z.B. ob mehr Schiffe verfehlt wurden als leere Felder vorhanden sind (u.ä.). Leider ist die Zielroutine nicht besonders ausgereift, da aber der Rechner für komplizierte Fallunterscheidungen einfach zu langsam ist, durchaus ausreichend. Der Aufbau ist variabel gestaltet. In Zeile 30 kann jeder individuell in die Variablen A(1)..A(4) = A..D seine gewünschte Anzahl Schiffe einsetzen. Alle weiteren Werte werden auf dieser Basis errechnet. Voraussetzung ist allerdings immer ein 10*10er Feld. Wer natürlich mehr Schiffe als Felder benutzt, darf sich nicht wundern, wenn der PC 2 Jahre für den Aufbau benötigt. Der Inhalt des Feldes ist bitorientiert, um bei kleinem Speicherbedarf möglichst viel Informationen aufbewahren zu können und weil für das Speichern nur die ASCII-Werte 32-95 zur Verfügung stehen. (Alle anderen Werte ausser 0, 251,252 lassen sich ja nicht mit den Funktionen 'ASC' und 'CHR\$' auslesen und umwandeln.) In Bit 0,1 steht die Bauteilnummer des Schiffes, in Bit 2, 3 die Art desselben und in Bit 4 steht, ob das Schiff horizontal oder vertikal liegt. Bit 5 und 6 zeigen an, ob an der Stelle Wasser oder ein Schiff ist. Der Aufbau der PC-Schiffe steht im Feld E\$(), den Merktzettel für die Schüsse des Rechners bildet das Feld F\$(). Ansonsten wird nur die Variable BB\$ für den Kontrollausdruck eingerichtet, alles andere sind Standardvariablen.

Listing:

```

10 REM SCHIFFE VERSENKEN, DIRK EVERLING,
   8.12.87, 05066/3759
20 "A" CLEAR: DIM E$(9,9)*1, F$(9,9)*1:
   PAUSE " SHIP DOWN": F$="0": E=0:
   INPUT "STARTZAHL: "; Z: Z=RND Z: Z=0
30 A(4)=1: A(3)=2: A(2)=3: A(1)=2:
   S=A(4)*4+A(3)*3+A(2)*2+A(1): T=S: U=100-S
40 FOR I=0 TO 9: FOR J=0 TO 9: E$(I,J)="u": F$(I,J)="0":
   NEXT J: NEXT I

```

```

50 GOSUB "AUFBAU":BEEP 3:PRINT "AUFBAU FERTICH !":L=0
60 "B"Q=RND(10)-1:R=RND(10)-1:I=Q:J=R
70 "START" GOSUB "KONTROLLE":INPUT "KOORDINATEN: ";P$:
  P$=LEFT$(P$,2)
80 V=VAL(LEFT$(P$,1)):W=VAL(RIGHT$(P$,1)):
  O=ASC(E$(V,W))
90 IF(O AND 96)=64 PRINT " ";V;W;"= WASSER !":
  GOTO "ANGRIFF"
100 BEEP 1:PRINT " ";V;W;"= TREFFER !":T=T-1:
  E$(V,W)=CHR$( (O AND 223) OR 64 )
110 Z$="0":N=(O AND 12)/4+1:K=(O AND 3)+1:
  L=(O AND 16)/16
120 ON N GOSUB "1","234","234","234":GOTO "START"
130 "ANGRIFF" IF F$="1" GOSUB "WEITER"
140 IF F$="0" LET I=Q:J=R:GOSUB "STRATEGIE"
150 PRINT " ZIEL: ";I;J:P$="X":INPUT " ERGEBNIS ?: ";P$:
  P$=LEFT$(P$,1)
160 IF P$<>"V" AND P$<>"T" LET P$=" ":F$(I,J)="1":E=E+1
170 IF E>U BEEP 3:CLEAR:PRINT "   MOGLER !"
180 IF P$=" " GOTO "START"
190 IF P$="V" LET F$="0":L=L+1:IF L>(A+B+C+D)BEEP 3:
  CLEAR:PRINT "   MOGLER !"
200 IF P$="T" AND F$="0" LET X=I:Y=J:F$="1"
210 F$(I,J)="2":S=S-1:ON S+1 GOSUB "KONTROLLE":
  GOTO "ANGRIFF"
220 "WEITER" IF X>9 IF F$(X+1,Y)="0" LET I=X+1:J=Y:
  GOTO "FERTIG"
230 IF Y<9 IF F$(X,Y+1)="0" LET I=X:J=Y+1:GOTO "FERTIG"
240 IF X>0 IF F$(X-1,Y)="0" LET I=X-1:J=Y:GOTO "FERTIG"
250 IF Y>0 IF F$(X,Y-1)="0" LET I=X:J=Y-1:GOTO "FERTIG"
260 IF X<8 IF F$(X+2,Y)="0" IF F$(X+1,Y)="2" LET I=X+2:
  J=Y:GOTO "FERTIG"
270 IF Y<8 IF F$(X,Y+2)="0" IF F$(X,Y+1)="2" LET I=X:
  J=Y+2:GOTO "FERTIG"
280 IF X>1 IF F$(X-2,Y)="0" IF F$(X-1,Y)="2" LET I=X-2:
  J=Y:GOTO "FERTIG"
290 IF Y>1 IF F$(X,Y-2)="0" IF F$(X,Y-1)="2" LET I=X:
  J=Y-2:GOTO "FERTIG"
300 IF X<7 IF F$(X+3,Y)="0" IF F$(X+2,Y)="2"
  IF F$(X+1,Y)="2" LET I=X+3:J=Y:GOTO "FERTIG"
310 IF Y<7 IF F$(X,Y+3)="0" IF F$(X,Y+2)="2"
  IF F$(X,Y+1)="2" LET I=X:J=Y+3:GOTO "FERTIG"
320 IF X>2 IF F$(X-3,Y)="0" IF F$(X-2,Y)="2"
  IF F$(X-1,Y)="2" LET I=X-3:J=Y:GOTO "FERTIG"
330 IF Y>2 IF F$(X,Y-3)="0" IF F$(X,Y-2)="2"
  IF F$(X,Y-1)="2" LET I=X:J=Y-3:GOTO "FERTIG"
340 "LOOP" I=RND(10)-1:J=RND(10)-1
350 IF((I=X)AND(J=Y))OR((I<>X)AND(J<>Y))OR F$(I,J)<>"0"
  GOTO "LOOP"
360 "FERTIG" RETURN
370 "STRATEGIE" I=I+2:IF I>9 LET J=J+1:
  I=I-9-2*(I-(INT(I/2))*2)
390 IF J>9 LET J=J-10-2*(J-(INT(J/2))*2):I=RND(2)-1
400 IF F$(I,J)>"0" GOTO "STRATEGIE"
410 Q=I:R=J:RETURN
420 "234" IF L=1 GOTO "HORI"
430 FOR M=(V-K+1)TO(V+N-K):IF((ASC(E$(M,W))AND 96)=32)
  LET M=V+N-K:Z$="1"

```

```

440 NEXT M:IF Z$="1" GOTO "ABBRUCH"
450 GOTO "1"
460 "HORI" FOR M=W-K+1 TO W+N-K:IF(ASC E$(V,M)AND 96)=32
    LET M=W+N-K:Z$="1"
470 NEXT M:IF Z$="1" GOTO "ABBRUCH"
480 "1" BEEP 2:PRINT " VERSENKT ! "
490 "ABBRUCH" RETURN
500 "AUFBAU" FOR L=1 TO A(4)
510 "AUSWAHL4"R=RND(2):V=RND(7)-1:W=RND(7)-1:Z$="0":
    Q=0:IF R=2 GOTO "VERT4"
520 FOR I=V TO V+3:IF E$(I,W)<>"ü" LET Z$="1"
530 NEXT I:IF Z$="1" GOTO "AUSWAHL4"
540 FOR I=V TO V+3:O=44 OR Q:Q=Q+1:E$(I,W)=CHR$(O):
    NEXT I:GOTO "END4"
550 "VERT4" FOR I=W TO W+3:IF E$(V,I)<>"ü" LET Z$="1"
560 NEXT I:IF Z$="1" GOTO "AUSWAHL4"
570 FOR I=W TO W+3:O=60 OR Q:Q=Q+1:E$(V,I)=CHR$(O):NEXT I
580 "END4" NEXT L:FOR L=1 TO A(3)
590 "AUSWAHL3"R=RND(2):V=RND(8)-1:W=RND(8)-1:Z$="0":
    Q=0:IF R=2 GOTO "VERT3"
600 FOR I=V TO V+2:IF E$(I,W)<>"ü" LET Z$="1"
610 NEXT I:IF Z$="1" GOTO "AUSWAHL3"
620 FOR I=V TO V+2:O=40 OR Q:Q=Q+1:E$(I,W)=CHR$(O):
    NEXT I:GOTO "END3"
630 "VERT3" FOR I=W TO W+2:IF E$(V,I)<>"ü" LET Z$="1"
640 NEXT I:IF Z$="1" GOTO "AUSWAHL3"
650 FOR I=W TO W+2:O=56 OR Q:Q=Q+1:E$(V,I)=CHR$(O):NEXT I
660 "END3" NEXT L:FOR L=1 TO A(2)
670 "AUSWAHL2"R=RND(2):V=RND(9)-1:W=RND(9)-1:IF R=2
    GOTO "VERT2"
680 IF E$(V,W)="ü" AND E$(V+1,W)="ü" LET E$(V,W)="$":
    E$(V+1,W)="%":GOTO "END2"
690 GOTO "AUSWAHL2"
700 "VERT2" IF E$(V,W)="ü" AND E$(V,W+1)="ü"
    LET E$(V,W)="4":E$(V,W+1)="5":GOTO "END2"
710 GOTO "AUSWAHL2"
720 "END2" NEXT L:FOR L=1 TO A(1)
730 "AUSWAHL1"V=RND(10)-1:W=RND(10)-1:IF E$(V,W)<>"ü"
    GOTO "AUSWAHL1"
740 E$(V,W)="- ":NEXT L:RETURN
750 "KONTROLLE" IF S<=0 PRINT " * GAME OVER * ":END
760 IF T<=0 PRINT " * GEWONNEN * ":END
770 RETURN
780 "M"BB$="":LPRINT " 0123456789":FOR W=0 TO 9:
    FOR V=0 TO 9:O=ASC(E$(W,V))
790 IF O=64 LET BB$=BB$+"+"
800 IF O<>64 LET BB$=BB$+STR$((O AND 12)/4+1)
810 NEXT V:LPRINT STR$(W);";";BB$;";":BB$="":NEXT W:END

```

Airplane

Beschreibung:

Die Aufgabe des Spielers ist es, ein Flugzeug durch ein Labyrinth zu fliegen, ohne dieses zu berühren. In einem Durchlauf darf das Flugzeug 100 mal Decke oder Boden berühren. Zu Beginn des Spiels hat der Pilot 3 Flugzeuge zur Verfügung. Für jeweils 1000 Punkte bekommt er ein zusätzliches. Das Flugzeug wird mit der <=> Taste (runter) und mit der <M+> Taste (hoch) gesteuert.

Das Spiel läuft nur auf PC 1403 (alle Speicherausbauten).

Wenn das Programm erstmalig gestartet wird dauert es 75 Sekunden bis das Maschinenprogramm im Speicher steht. Beim erneuten Start durch <run> steht das Programm sofort zur Verfügung.

Das Programm benötigt 2675 Bytes für Maschinensprache und Labyrinthdaten, deshalb ist es möglichst kurz geschrieben, so daß es auch auf der 8 k Version läuft.

PS.: Wer das Flugzeug nicht direkt auf dem Display findet, es ist an der Stelle, wo sich normalerweise der dritte Buchstabe befindet.

Listing:

```

1  IF M=7 GOTO 5
2  DIM C$(0)*80:FOR II=64272 TO 64420 STEP 37:
   READ C$(0):FOR B=0 TO 36
3  POKE II+B,16*(ASC MID$(C$(0),2*B+1,1)-65)+ASC
   MID$(C$(0),2*B+2,1)-65:NEXT B:NEXT II:CALL 64272
4  CLEAR:FOR I=0 TO 119:POKE &FA00+I,PEEK(&F200+I):
   NEXT I:M=7
5  Q=999:P=3:S=0:POKE &FAF7,0,&F2:CALL 61599
6  POKE &FAF9,0,0,0:PAUSE "PLANES "+STR$ P+"
   SCORE "+STR$ S:P=P-1:CALL 61610:S=S+PEEK &
   FAF7+256*PEEK &FAF9
7  IF S>Q LET P=P+1:LET Q=Q+1000:GOTO 7
8  IF S>H LET H=S
9  IF P GOTO 6
10 PRINT "SCORE "+STR$ S+" HIGH "+STR$ H:
   REM (C) G.E.KOLLES
11 "0?20
12 "4?ak0P=B4OI3DSOdJ6Lm>0MW;1PF;2DüteIGIj0n7M5JDQ7
   kEs5JDRfB4?a1E1L2>0aAjU8AsEOA
13 "jU8üteMGQ=bAt5MülfüüteQCDA7mDfDüteQGIuXX10;hDXH
   2NMcu0eaKDüD792ISr2T54?akEnSK
14 "4HQGgSP84OUGüU8b=g71EsbAREO6>0PAtEM2DRfV4HYGgSP
   84OUGüU8b6A6;ErHn217oEj9B;0fA
15 "S5O6>0HAtEM2DQ7kEnSK4HQGAscB4OIG1g69EjOKDQ71EsbA
   RUM71k8AsUOK4HcGAscB4OIG1g6<
16 "EjOKDdBnQ600HGUüH01QtPEüH01üH01Q<54792IS7BT5QV0
   0HBj792IS>hT5QV00HAn792IS;2T5
17 "QV00Hff692MSGRT5QV00HGb692MSKBT5QV00HEj692MSü2T
   54?apEnCK40cGD=b4lg3ptk:4lg7p
18 "DQ3ptEMWI7ifZSM0ü410ü410ü410ü41PH61PH61PH61PH6
   1PH61PH61fL71fL71fL71fL71fL71f

```

- 19 "L71fL7QnN7QnN7QnN7QnN7QnN7QnO7grO7grO7grO7grO7grO
7gnN7QnN7QnNGUoLG5gLG5gLm=iHl=S
- 20 "Hl=WI1MWAjM7AjM?Cfr?3fr?3fr?7grO7grO7g0F5QrO41Hü
7grü6QPO7gfA71rO41XD7grü7grü
- 21 "5Q8O7g0K41rO7Q0N7grO7grO41HB7grG7grü4QH07gLO7g0K
51rO41Hü7grü5gLO7g0G5grO418F
- 22 "7grG5AXN7grO7grO7grO7fr?1fL3üj5QH71fN7QrO7grO
7grN7QfL65Qüj<71fr?7grO7grO3fr?
- 23 "1f<30D51üF5QHG5gLGUoNGUoNGUgLG5gLG=iLm=iHl=SH1=WI
1MWI1MWI1MSH1=SH15QHF5QLG5g
- 24 "LGUoNGUoNGUoLG5iLm=iLm=iLm=gLG5gLG5gLG5oNGUoNGU
oNGUgHF5QUd=3üjM7Ajs?Cjs?Cjs?
- 25 "CjM7üj=1üD51HF5QLG5gLG5gLG5gLG5gLG=iLm=iLm=iLG
5gLG5gLG5gNGUoNGUoNG5gLG5gLG5g
- 26 "LG5gLG5gLG=SH1=Süj=3üj=3AjM7AjM7AjM7Aj=3üj=3ü1
5QLG5gLG5gLF5QüD51üD51üD53ü1=S
- 27 "Hl=SH1=SH1MWI1MWI1MWI1MSH1=SH1=SH1=SH1=iLm=iLm=
iLl=SH1=SH1MWI1MWIjM7AjM7Cjs?
- 28 "Cjs?Cjs?Cjs?Cjs?Cjs7üj=3üj=1üD51HF5QL71fN7QnN7gr
O7grN7QnN7QfLG5gLF5QHD510f<3
- 29 "0fL73fr?3fr?7grO7grO7grO7fr?1fL71f<3üj=QHG5fL
7QnN7QnN71fL65QHF5QHD53üj<30f<3
- 30 "1fL?3fr?7grO7grO7grO7grO3fr71f<3üD5PH71fN7QrO
7grO7QfH4000ü<73grO7grO3fL30ü00
- 31 "0010H71nO7grO7gnL61000000ü<73grO7grO7fr70f410ü5
1üD51üF5gNGUoNGUgHD51üD53Ajs?
- 32 "Cjs?Aj=1üD51üD51HG5oNGUoNGUgHD51üD=3üj=SH1=SH1=
SH1=SH1=SI1MWI1MWI1=SH1=SH1=S
- 33 "Hl=SH1=SH1=SH1=SLm=iLm=iLm=SH1=3üjL71fr?3frO7gr
O7fr71fL70f<10D10ü61PH71fN7gr
- 34 "O7grO7grO7grN7QnN7QnL71fLG5gLG=iLm=iLm=gLG5gLG
5gHF5QHF5QHF=SH1=SH1=SI1MWI1MW
- 35 "IjM7AjM7Cjs?Cjs?3fr?3fr?3grO7grO7grO7grO7grO3fr?
3fL71fL30j=3üF5QHG5gNGUoNGUo
- 36 "NGQnN7QnN7QnO7grO7grO7QnN7QnN7QnN7QfL71fL71fL6
1PH61PH610ü451üD51üF5QHF=SLm=i
- 37 "Lm=iLm=iLl=QHF5QHF5QHF5QHF=SH1=SI1MWI1MWIjM7üj=
3üD51üD51üD51HF5QHF1PH61PL71f
- 38 "L71fL71fL7QnN7QnN7QrO7grO7grO7QnN7QnN71fL71PHF
5QHF5SH1=SüjM7AjM7AjM?Cfr?3fr?
- 39 "3grO7grO7grO7grO3fr?3fr?3jM7AjM7AjMSH1=SH1=QLG
5gLG5gLG5gLG5gLF5QHF5Qüj=3üjM7
- 40 "AjM7Cjs?Cjs?Cjs?CjM7AjM7AjM7AjMSH1=SH1=SH1=SHF
5QHF5gLG5gLG5gLG5gHF51üD510ü41
- 41 "0f<71fr?3frO7grO7grO7grO7grO3fr?1f<30f41üD10H6
1PL71nN7QnN7QnN7grO7grO7grN71P
- 42 "u410üD=7Cjs?Cjs?Cjs?Cjs?Cjs?Cfr?3grO7grO7grO7gr?
1fL30ü41üF5gNGUoNGUoN7QnN7Qn
- 43 "N7grO7grO7grN7QnN71fL71PH61QüD51üf<30fL71fL?3fr?
7grO7grO7grO7grO7fr?1f<30ü40
- 44 "u41PH71fN7QrO7grO7grN7QnN71fLG5gLG5gLG5gNGUoNGU
oNGUoNG5gLG5gLm=iLm=iLm=iLm5g
- 45 "LG5gHF5QHF5QüD53üj=3AjL71fr?3fr?7grO7grO7fr?1fL7
1fL70j=3üj=3ü1=SHF5QHG5gLG5g
- 46 "LG5gLG1nN7QnN7QnO7grO7grO7grN7QfL61PüD430fL73frO
7grO7fr?1fL30f41ü41PH71fN7Qr
- 47 "O7grO7QnN71fL61QHD53üf<71fL?3frO7grO7grO7grO7gr?
3fr71fL30j=1üD50H61PH61fL71f

48 "N7QnN7gr07gr07QnN7QnN7Qr07gr07gr07QnN7QfL71fL6
1PH61QüD51üD<30f<71fL71fr?3fr?
49 "7gr07gr07gr07gr?3fr?3fr?1fL71jM7Aj=3üj=3üj=3ül=
SHm=iLm=iLm=iHl=SHl=SHl=WI1MW
50 "IjM7AjM7AjM7Cjr?3fr?3fr07gr07gr03fr?3fr?1fL71f=
3üj=1HF5QH71fL7QnN7Qr07gr07gr
51 "07QnN7QnN7QfL71fLF5QHF5QüD53üj<30fL71fL?3fr?3frO
7gr07gr?3fr?3fr?3fL71fL30f<3
52 "0j=3üj=3üj=3üj=3Hl=SHl=SHG5gLG5gLG5gLG5gLF5QHF=
SHl=SHl=SHl=SHl=SHl=Süj=3üj=3
53 "üj=3üj=3üj=3üj=3üj=7AjM7Cjs?Cjs?Cjs?Cjs?Cjs7AjM
7AjM7ül-SHl5QHF5PH71fL71fL71f
54 "L61PH61PH000z
55 DATA "IEACDANLIFACIANLAFAFAFCEGHEPCJAECECEGHEMC
JAMCECEGHFDCJBEAEHIPLLKHIPLGDIH
56 DATA "BAPLNJFHNLIGBBNKFHNLAHAEHIPLLKCEGHKDIKGBHA
NDJALAFHIPLGDBAPLNIFHCGBAPLNJFH
57 DATA "CGBAPLNKFHCGCNBNDHHIPLLPBAPLNIFCHIPLLPBAPLN
JFCHIPLLPBAPLNKFCHIPLLPBAPLNLFC
58 DATA "IKBBNIFHFKFKEGPMNLBBNJFHFIGEADENLBBNIFCBBN
JFHFIGEPANLBBNKFHNCNCGEAPENLBB
59 DATA "NJFCBBNKFHNCNCGEMANLBBNLFHGEDPEHNLBBNKFCD
HAEAEAEAEDHCEGHGBDKADHFAGHFDADH

Kapitel 4.5

Sonstige Fachgebiete

Biorythmus

Beschreibung:

Das Programm berechnet anhand von Sinusfunktionen, mit verschiedenen Periodendauern, Ihren augenblicklichen Zustand für das Gefühl, den Geist sowie auch für den Körper. Die Werte können für einen Tag im Display, sowie für einen ganzen Monat auf dem Drucker dargestellt werden. Die Sinuskurven beginnen am Tag Ihrer Geburt.

Funktionsweise:

Sie werden nach dem Programmstart aufgefordert, Ihre Geburtsdaten einzugeben. Zuerst den Tag, danach den Monat und zum Schluß das Jahr. Jetzt zeigt Ihnen der Rechner den Wochentag Ihrer Geburt an. Als Eingabedatum müssen Sie das Datum des Tages eintippen, für den Tag die Werte berechnet werden sollen. Es wird Ihnen wiederum der Wochentag des Eingabedatums angezeigt. Kurz danach erscheint im Display "INT:XX.XX%" (Geist). Nach «ENTER» erscheint "EMO:XX.XX%" (Gefühl) und drückt man jetzt nochmals die «ENTER»-Taste wird "VIT:XX.XX%" (Körper) im Display sichtbar. Drückt man die «ENTER»-Taste nun zum 3. Mal, so erscheint immer abwechselnd: "KURVE" - "TABELLE" - "NEUEINGABE" - "KURVE" - "TABELLE"..... Drücken Sie beim Erscheinen des Gewünschten eine beliebige Buchstaben- oder Zifferntaste für ca. 1 Sekunde, und der Rechner wird diese Funktion sofort ausführen. Nach der Ausführung wird wieder ins Menü gesprungen, so daß es möglich ist, Kurve und Tabelle hintereinander auszudrucken.

KURVE:

Es werden die 3 verschiedenen Sinuskurven für jeweils den ganzen Monat des Eingabedatums ausgedruckt. Danach wird ins Menü zurückgesprungen.

TABELLE:

Es werden ab dem Eingabedatum die jeweiligen 3 Werte für einen Tag bis zum Ende des Monats in tabellarischer Form ausgedruckt. Möchte man nun einen ganzen Monat ausdrucken, so muß man als Eingabedatum den 01. dieses Monats eingeben. Dies wurde so gewählt, um Thermopapier zu sparen. Nach der Ausführung wird automatisch ins Menü gesprungen.

NEUEINGABE:

Der Rechner springt zum Programmanfang.

PROGRAMMENDE:

Das Programm wird mit «BRK» beendet.

<u>Beispiel:</u>	Rechner:	Eingabe:
	"GEBURTSDATUM?"	
	"TAG:"	26 «ENTER»
	"MONAT:"	1 «ENTER»
	"JAHR:"	62 «ENTER»
	"FR 26.01.62"	
	"EINGABEDATUM:"	
	"TAG:"	17 «ENTER»
	"MONAT:"	8 «ENTER»
	"JAHR:"	84 «ENTER»
	"FR 17.08.84"	
	"INT: -86.60%"	«ENTER»
	"EMO: 100.00%"	«ENTER»
	"VIT: 97.91%"	«ENTER»
	"KURVE"	
	"TABELLE"	
	"NEUEINGABE"	
	"KURVE"	«SPC»
	Drucker druckt KURVE fuer Aug. 84 aus. Danach:	
	"KURVE"	
	"TABELLE"	«SPC»
	Drucker druckt TABELLE ab 17.08. aus. Danach:	
	"KURVE"	
	"TABELLE"	
	"NEUEINGABE"	«SPC»

Listing:

```

20 DIM L$(1)*25: DIM O$(3)*19
30 "N" PAUSE " GEBURTSDATUM ?": RADIAN
40 GOSUB 150:A=N:Q$=H$:W$=J$:R=Y
50 PAUSE "HEUTIGES DATUM ?": GOSUB 150:N=N-A
60 GOSUB "I": GOSUB "E": GOSUB "V"
70 PRINT " INT: ";I;"%": PRINT " EMO: ";E;"%":
PRINT " VIT: ";V;"%"
80 PAUSE " KURVE": IF INKEY$ <>" " THEN "K"
90 PAUSE " TABELLE": IF INKEY$ <>" " THEN "T"
100 PAUSE " NEUEINGABE": IF INKEY$ <>" " THEN "N"
110 GOTO 80
120 "I" I=((N/33)- INT (N/33))*2* PI :I= INT (.5+10000
* SIN I)/100:U=I: RETURN
130 "E" E=((N/28)- INT (N/28))*2* PI :E= INT (.5+10000
* SIN E)/100:U=E: RETURN
140 "V" V=((N/23)- INT (N/23))*2* PI :V= INT (.5+10000
* SIN V)/100:U=V: RETURN
150 INPUT " TAG: ";D: INPUT " MONAT: ";M:
INPUT " JAHR: ";Y: IF Y<100 THEN LET Y=Y+1900
160 N=Y*365+ INT (Y/4)+ INT (Y/400)- INT (Y/100): IF
INT (Y/4)<>Y/4 AND M=2 AND D>28 THEN
PRINT "FALSCHDATUM !": GOTO "N"
170 IF INT (Y/4)=Y/4 AND M<3 THEN LET N=N-1
175 L$(1)="231223345566":N=N-20+ ASC MID$
(L$(1),M,1)+30*(M-2)+D
180 P=N-7* INT (N/7)+1:L$(1)="SASOMODIMIDOFR"
185 B$= MID$( L$(1),2*P-1,2):X=D: GOSUB 190:
PAUSE " ";B$;" ";H$;" ";J$;" ";K$: RETURN
190 H$= STR$ X: IF X<10 THEN LET H$="0"+H$
200 J$= STR$ M: IF M<10 THEN LET J$="0"+J$

```


<u>Beispiel:</u>	Rechner:	Eingabe:
	"GROESSE IN CM:"	175
	"GEWICHT IN KG:"	70
	"KOERPEROBERFL.:"	
	"1.848 QUADRATM."	

Listing:

```

10 INPUT "GROESSE IN CM :";H
20 INPUT "GEWICHT IN KG :";W
30 BSA=H^.725*W^.425*71.84E-4
40 IF BSA<.6 THEN 60
50 PAUSE "KOERPEROBERFL. :";
   USING "##.###&&&&&&&&&": PRINT BSA;" QUADRATM.":
   USING : GOTO 10
60 W=W*1000:BSA=H^.3*W^(.7285-.0188* LOG W)*3.207*10^-4
   : GOTO 50

```

Stundenplan

Beschreibung:

Dieses Programm ermöglicht Schülern und Studenten nach der Eingabe des Datums und der Uhrzeit zu erfahren, in welchem Unterrichtsfach und in welchem Zimmer sie an diesem Tag und zu dieser Zeit unterrichtet werden. Ferien und schul- bzw. vorlesungsfreie Tage werden vom Rechner natürlich nicht berücksichtigt.

Funktionsweise:

Vor dem Programmstart müssen Sie Ihren kompletten Stunden- bzw. Vorlesungsplan in DATA-Zeilen abspeichern. Dies geschieht in folgender Reihenfolge:

1. Wochentag 2.Stundenanfang 3.Stundenende 4.Fach/Zimmer

Beispiel:

```

200 DATA "MO",8.00,9.30,"MATHEMATIK 109","MO", 9.30, 9.45, " PAU-
SE",... Sind alle Stunden eingetragen, so wird am Schluß der letzten DATA-
Zeile ,"ENDE" eingegeben. Wurde alles ordnungsgemäß ausgeführt, so kann
das Programm gestartet werden. Zuerst werden Sie nach dem Datum (Bsp.
17.08.84) gefragt, wobei Sie bei "TAG:" 17, bei "MONAT:" 8 und bei "JAHR:"
84 eintippen. Danach werden Sie vom Rechner aufgefordert, die gewünschte
Uhrzeit einzutippen z.B. 11.15. Nach kurzer Rechenzeit wird Ihnen das Fach
und das Zimmer angezeigt werden. Nach «ENTER» erscheint dann wieder
"TAG:". Möchten Sie nun dasselbe Datum noch einmal benutzen, so brauchen
Sie nur «ENTER» zu drücken und Sie werden nach der Uhrzeit gefragt. Sollte
das Programm nicht ordnungsgemäß funktionieren, so ist mit höchster Wahr-
scheinlichkeit ein Fehler in einer DATA-Zeile gemacht worden. Bitte über-
prüfen Sie dann die DATA-Zeilen im Programm.

```

Beispiel:	Rechner:	Eingabe:
	"TAG:"	17 «ENTER»
	"MONAT:"	8 «ENTER»
	"JAHR:"	84 «ENTER»
	"UHRZEIT:"	10.00 «ENTER»
	"MATHEMATIK 222"	«ENTER»
	"TAG:"	«ENTER»
	"UHRZEIT:"	20.00 «ENTER»
	"FREIZEIT"	

Listing:

```

10 DIM L$(2)*25
20 PAUSE "* STUNDENPLAN *": PAUSE "** PROGRAMM **"
30 INPUT "TAG: ";D: INPUT "MONAT: ";M:
   INPUT "JAHR: ";Y: IF Y<100 LET Y=Y+1900
40 N=Y*365+INT(Y/4)+INT(Y/400)-INT(Y/100):
   IF INT(Y/4)=-Y/4 AND M<2 AND D<28 THEN
   PRINT "FALSCHDATUM!": GOTO 30
50 IF INT(Y/4)=Y/4 AND M<3 LET N=N-1
60 L$(1)="2312233445566":N=N-20+ASC MID$(
   (L$(1),M,1)+30*(M-2)+D
70 P=N-7*INT(N/7)+1:L$(1)="SASOMODIMIDOFR"
80 B$=MID$(L$(1),2*P-1,2)
90 INPUT "UHRZEIT: ";U: RESTORE
100 READ A$: IF A$="ENDE" THEN 130
110 READ A1,A2,L$(2): IF A$=B$ AND A1<=U AND A2>U THEN
   PRINT L$(2): GOTO 30
120 GOTO 100
130 PRINT "FREIZEIT!": GOTO 30
200 DATA "MO",8.00,9.30,"MATHEMATIK 109","MO",9.30
   ,9.45,"PAUSE","MO",9.45,11.15
210 DATA "PHYSIK 201","MO",11.15,11.30,"PAUSE"
   ,"MO",11.30,13.00,"DEUTSCH 111"
220 DATA "DI",8.00,9.30,"INFORMATIK 222","DI",9.30
   ,9.45,"PAUSE","DI",9.45,11.15
230 DATA "ENGLISCH 203","DI",11.15,11.30,"PAUSE"
   ,"DI",11.30,13.00,"MATHEMATIK 109"
240 DATA "MI",9.45,11.15,"ENGLISCH 119","MI",11.15
   ,11.30,"PAUSE","MI",11.30,13.00
250 DATA "GESCHICHTE","DO",8.00,9.30,"PHYSIK","DO",9.30
   ,9.45,"PAUSE","DO",9.45
260 DATA 11.15,"DEUTSCH 224","DO",11.15,11.30,
   "PAUSE","DO",11.30,13.00,"SPORT 021"
270 DATA "DO",13.00,14.00,"MITTAG","DO",14.00,15.30
   ,"CHEMIE 009","FR",8.00,9.30
280 DATA "DEUTSCH 123","FR",9.30,9.45,"PAUSE","FR"
   ,9.45,11.15,"MATHEMATIK 222"
290 DATA "FR",11.15,11.30,"PAUSE","FR",11.30,13.00,
   "ENGLISCH 213","ENDE"

```

Großschrift mit CE 126

Beschreibung:

Dieses Programm ermöglicht es, Texte mit 12 verschiedenen Zeichensätzen auf dem Drucker CE-126 auszudrucken. 4 verschiedene Druckbreiten und 3 verschiedene Druckhöhen sind möglich, wobei sich jede Breite mit jeder Höhe kombinieren läßt. Bis auf zwei Ausnahmen (Pi u. Wurzelzeichen), lassen sich alle Buchstaben, Ziffern und Sonderzeichen im Längsformat ausdrucken.

Um die Länge des Programms möglichst gering zu halten, wird der Zeichensatz des PC 1401 als Zeichengenerator benutzt, d.h. die Zeichen sind so aufgebaut, wie sie normalerweise im Display dargestellt werden.

Funktionsweise:

Nach «RUN» erscheint im Display "GROSS-SCHRIFT" "MIT DEM CE-126" "TEXT EINGEBEN:". Nun den gewünschten Text, max. 80 Zeichen, eingeben. Nach «ENTER» blinkt dann "BREITE 1/2/3/4". Durch das Drücken auf eine der vier Zifferntasten stellt man die gewünschte Schriftbreite ein. Nach (BEEP) blinkt dann "HOEHE 1/2/3", wobei man hier durch das Drücken der «1», «2» oder «3» die gewünschte Schrifthöhe wählen kann. Kurz nach (BEEP) beginnt der Ausdruck. Ist der Ausdruck beendet, so zeigt der Rechner wieder "TEXT EINGEBEN:" an. Nun kann man einen neuen Text eingeben oder man kann das Programm beenden, indem man einfach "AUS" «ENTER» eintippt - der Rechner schaltet sich ab.

<u>Beispiel:</u>	Rechner:	Eingabe:
	"GROSS-SCHRIFT"	
	"MIT DEM CE-126"	
	"TEXT EINGEBEN:"	ABCDE «ENTER»
	"BREITE 1/2/3/4"	
	"BREITE 1/2/3/4"	1
	(BEEP)	
	"HOEHE 1/2/3"	1
	(BEEP)	

Nun beginnt der Ausdruck. Nach dem Ausdruck:

"TEXT EINGEBEN:" AUS «ENTER»

Rechner hat abgeschaltet.

Listing:

A C H T U N G : Zeile 1 unbedingt mit eingeben ! Hier wird ein Maschinenspracheprogramm hineingepokt.

```

1 REM ABCDEFGHIJKLMNOPQRSTUVWXYZ
5 RESTORE : FOR N=1 TO 8 : READ V : NEXT N :
  FOR N=14341 TO 14362 : READ V : POKE N,V : NEXT N

```

```

10 DIM A$(1)*80: DIM C(5): P=14359: Q=14358:
   PAUSE "GROSS-SCHRIFT": PAUSE "MIT DEM CE-126"
20 INPUT "TEXT EINGEBEN: "; A$(1): IF A$(1)="AUS"
   THEN CALL &05B0
25 WAIT 15
26 PRINT "BREITE 1/2/3/4": Z$= INKEY$: IF VAL (Z$)<1
   OR VAL (Z$)>4 THEN 26
27 BEEP 1
28 PRINT " HOEHE 1/2/3": Y$= INKEY$: IF VAL Y$<1 OR
   VAL Y$>3 THEN 28
29 BEEP 1: LPRINT ""
30 FOR K=1 TO LEN (A$(1)): B= ASC ( MID$( A$(1), K, 1)) * 5 +
   32740: IF B=32900 THEN FOR N=1 TO VAL Z$*5+3:
   LPRINT "": NEXT N: NEXT K: GOTO 20
40 FOR L=5 TO 1 STEP -1: C(L)= PEEK ( B-L): NEXT L
50 FOR L=1 TO 5: X=C(L): FOR M=1 TO VAL (Z$): IF M>1
   THEN LET C(L)=X
60 CALL 14341: RESTORE
70 READ D: IF D=0 THEN 110
80 E=C(L)/D: IF E<-1 THEN LET G=127: GOSUB 130: GOTO 100
90 IF E<1 THEN LET G=32: GOSUB 130: GOTO 70
100 C(L)=(E-INT (E))*D: GOTO 70
110 FOR N=1 TO INT ((24 VAL Y$*7)/2)+1: POKE P, 32: CALL Q
   : NEXT N: CALL 43396
120 NEXT M: NEXT L: LPRINT "": LPRINT "": LPRINT "":
   NEXT K: GOTO 20
130 FOR N=1 TO VAL Y$: POKE P, G: CALL Q: NEXT N: RETURN
140 DATA 64, 32, 16, 8, 4, 2, 1, 0, 16, 70, 215, 214, 36, 40, 3, 235,
   209, 16, 96, 186, 132, 27, 121, 128
150 DATA 69, 2, 0, 121, 167, 250

```

WICHTIG: Besitzer eines PC 1402 müssen folgende
 Änderungen am Programm vornehmen:

```

Zeile 5   N=8197 TO 8218
Zeile 10  P=8215 : Q=8214
Zeile 60  CALL 8197
Zeile 110 CALL 43396

```

Sollten Sie noch einen Rechner mit einem alten ROM besitzen,
so müssen Sie folgende Änderungen am Programm vornehmen:

```

Zeile 5   FOR N=14341 TO 14362
Zeile 10  P=14359 : Q=14358
Zeile 60  CALL 14341
Zeile 110 CALL 43296
Zeile 150 DATA 69,2,0,121,167,150

```

Namenserkenkung

Beschreibung:

Dieses Programm ist ein schönes Beispiel für sogenannte "Künstliche Intelligenz". Es ermittelt nach der Eingabe eines Namens, ob es sich bei dem Namen um einen weiblichen oder männlichen Vornamen handelt. Die Leistungsfähigkeit eines Taschencomputers läßt sich mit diesem Programm eindrucksvoll demonstrieren. Andere Einsatzmöglichkeiten wären z.B. die automatische Einfügung der Anrede bei einem Programm zur Erfassung von persönlichen Daten, als Gag auf Partys usw.

Funktionsweise:

Nach dem Programmstart werden Sie aufgefordert, einen Namen einzugeben (nur deutsche Namen !). Noch etwas Rechenzeit und es wird Ihnen vom Rechner das zum Namen gehörige Geschlecht angezeigt.

<u>Beispiel:</u>	Rechner:	Eingabe:
	"NAME EINGEBEN"	DIETER
nach 15 sek.:	"DIETER"	
	"IST EIN MANN"	

Listing:

```

20 DIM NS(1)*20
30 INPUT "NAME EINGEBEN ";NS(1)
40 G=0: RESTORE
50 TS= RIGHTS (NS(1),2)
60 READ TT$,GG: IF TT$="ENDE" THEN 110
70 IF TS-TT$ THEN LET G=GG: GOTO 60
80 GOTO 60
110 READ TT$,GG: IF TT$ "ENDE" THEN GOTO 140
120 IF TT$=NS(1) THEN LET G=GG: GOTO 140
130 GOTO 110
140 BEEP 0: WAIT 80: IF G=1 THEN PRINT NS(1):
    PRINT " IST EIN MANN !"
150 IF G=2 THEN PRINT NS(1): PRINT " IST EINE FRAU !"
160 IF G=0 THEN PRINT NS(1): PRINT "NAME UNBEKANNT !"
170 GOTO 30
310 DATA "ER",1,"LD",1,"AN",1,"RD",1,"NZ",1,"UT",1,"AR"
    ,1,"ON",1,"AM",1,"DO",1
315 DATA "RT",1,"ES",1,"GE",2,"LY",2,"OF",1,"KO",1,"LF"
    ,1,"AD",1,"NO",1,"UD",2
320 DATA "DA",2,"SI",2,"IA",2,"NI",2,"TE",2,"IG",1,"DE"
    ,2,"ED",1,"NG",1,"TA",2
330 DATA "TO",1,"AX",1,"EX",1,"CH",1,"RE",2,"NS",1,"IT"
    ,1,"ST",1,"KA",2,"AR",1
340 DATA "OR",1,"EF",1,"EL",1,"EN",1,"NK",1,"IM",1,"PH"
    ,1,"AS",1,"HI",2,"YN",2
350 DATA "RA",2,"NE",2,"EE",2,"KE",2,"EA",2,"IC",1,"IE"
    ,2,"NA",2,"DO",1,"TL",2
360 DATA "JA",2,"CA",2,"GO",1,"LI",2,"RK",1,"NN",1,"TZ"
    ,1,"ND",1,"EV",1,"OT",1

```



```

370 DATA "LE",2,"LA",2,"IS",2,"VI",2,"IN",2,"TH",2,"GA"
,2,"TY",2,"SE",2,"DT",1
390 DATA "DI",2,"UL",1,"MY",2,"GI",1,"JO",1,"IL",1,"NY"
,1,"KI",1,"ID",2,"VE",2
400 DATA "LM",1,"RY",1,"HE",2,"OB",1,"JE",2,"HA",2,"LL"
,2,"IX",2,"AV",1,"RN",1
405 DATA "OE",1,"US",1
410 DATA "MA",2,"DO",1,"RG",1,"CK",1,"SA",2,"AI",1,"AY"
,1,"WE",1,"RL",1,"ENDE",0
420 DATA "CHARLIE",1,"DAGMAR",2,"WILLI",1,"EDELTRAUD"
,2,"WALTRAUD",2,"WILLY",1
425 DATA "DAVID",1,"KONSYANTIN",1,"CHRISTEL"
,2,"HILDEGARD",2,"RUDI",1,"HILBORG",2
430 DATA "BENE",1,"BORIS",1,"ERWIN",1,"MALI",2,"RUTH",2
,"HEINI",1,"AGNES",2
435 DATA "KARMEN",2,"ALOIS",1,"INES",2,"BAERBEL",2
440 DATA "BIRGIT",2,"MARION",2,"MARGIT",2,"INGEBORG"
,2,"ULI",1,"MARIN",1
445 DATA "KIRSTEN",2
450 DATA "ELLEN",2,"WALTRAUT",2,"HELMUTH",1,"TRUDEL",2
,"KAREN",2,"VALENTIN",1
460 DATA "ENDE",0

```

ACHTUNG: Eingegebenes Listing unbedingt auf Tippfehler überprüfen!

Mini-Menü

Beschreibung:

Bekanntlich hat der SHARP-PC 18 Tasten, über die Funktionen oder Programme mittels DEF aufgerufen werden können. Es gibt aber immer wieder Fälle, bei denen diese 18 Tasten nicht ausreichen. Doch dafür gibt es Abhilfe mit folgenden Programmzeilen.

Dabei werden in Zeile 2 nach ON A GOTO hintereinander die Zeilennummern der Anfangszeile der einzelnen Programme eingegeben (z.B. 1000,1500,2000 usw.). Wird nun das Programm gestartet, so erscheint 'WELCHES PGM?' und man gibt eine Zahl gefolgt von ENTER ein. Hätte man in diesem Beispiel eine 1 eingegeben, so würde das Programm, das ab Zeile 1000 steht gestartet, bei 2 das Programm ab 1500 usw. Auf diese Weise können bis zu ca. 20 Programme aufgerufen werden, d.h. Zahlen zwischen 1 und 20. Damit man nun weiß, welches Programm sich hinter welcher Nummer verbirgt, sollte man eine Liste auf ein Klebeetikett schreiben und dies dann in den Schutzdeckel des Rechners kleben!

Listing:

```

1 BEEP(1) : INPUT "WELCHES PGM ?";A
2 ON A GOTO 1000,1500,2000,5000,8970
3 BEEP(1) : PAUSE "FALSCH EINGABE" : GOTO 1

```

Telefongebührenzähler

Beschreibung:

Dieses Programm zeigt den Einsatz des SHARP-PC als zeitabhängigen Zähler. Es wurden alle wichtigen Tarife berücksichtigt. (Stand Mai 1984) Verwendete Tarife (ohne Gewähr)

Tarif 1 : 8 Minuten Takt
 Tarif 2 : Nachttarif 1min = 0,46 DM
 Tarif 3 : Tagtarif 1min = 1,15 DM
 Tarif 4 : Ausland meist 1min = 1,33 DM
 (nähere Auskunft beim Fernmeldeamt)

Die Tarife 2 bis 4 sind Ferngespräche. Das Programmieren weiterer Tarife ist in Zeile 1030 leicht zu realisieren. Dabei muß der Variable L die Kosten je Minute geteilt durch 35,4 zugewiesen werden.

Bei einzelnen Rechnern kann es möglich sein, daß die Zeit für den Ortstarif geeicht werden muß, da der Rechnerquarz eine zu hohe Toleranz hat. Dazu muß mit einer Stoppuhr die Zeit gemessen werden, wobei nach etwa 8 Minuten mehrere Piepstöne hörbar sein müssen. Stimmt die Zeit nicht, muß das WAIT-Intervall in Zeile 1112 (normal: 52) je nachdem geändert werden. Die Werte der Tarife 2 bis 4 werden in Zeile 1042 / 1044 / 1046 geändert.

Funktionsweise:

- Starten des Programmes mit DEF G
- Anzeige (TARIF: 1/2/3/4)
- Benötigten Tarif eingeben
- Anzeige " START ? "
- Bei Gesprächsbeginn ENTER drücken
- Bei Gesprächsende Taste H drücken
- Gebührenbetrag wird in DM auf der Anzeige dargestellt

Listing:

```

1010 "G" P=0:L=0:U$="DM":K$="##.##&&&": USING
1020 INPUT "TARIF: 1/2/3/4";F
1030 ON F GOTO 1040,1042,1044,1046
1040 GOTO 1110
1042 L=.013: GOTO 1050
1044 L=.315: GOTO 1050
1046 L=.36: GOTO 1050
1050 PRINT "START?": BEEP 1
1060 GOSUB 1090:P=P+L
1070 IF INKEY$ ="H" THEN 1100
1080 GOTO 1060
1090 WAIT 83: PRINT USING K$;U$;P
1092 RETURN
1100 USING : PAUSE "SUMME: ": WAIT : PRINT USING K$;U$;P:
    END
1110 FOR X=1 TO 475
1112 WAIT 52: PRINT "8 MINUTEN": NEXT X
1114 BEEP 9
1116 END

```

Digitaluhr

Beschreibung:

Hier eine kleine Digitaluhr mit Wecker für den PC. Sie läuft sogar relativ genau mit etwa 1 min Differenz am Tag. Allerdings schwankt dieser Wert von Rechner zu Rechner. Man muß also einen Feinabgleich der Zählschleife in Zeile 130 vornehmen. Es ist die Zahl 2750 entsprechend zu korrigieren, je nach dem, ob die Uhr zu schnell oder zu langsam ist.

Der besondere Kniff an diesem Programm liegt in der Erzeugung der Stunden und Minuten. Es wird nämlich der Winkel eines virtuellen Zeigers vom Gradmaß in die Stunden und Minuten mittels des DMS-Befehls umgewandelt. Somit kann ohne großen Programmaufwand eine 24h-Anzeige realisiert werden.

Listing:

```

50 "A" CLEAR 80 PRINT "DIGITALUHR"
100 USING "###.###"
105 INPUT "WECKEN J/N ?";D$: IF D$="J" INPUT "WECKZEIT?";C
110 "B" INPUT "UHRZEIT";A
120 B= DEG A
130 B=B+1/2750
135 IF B>24.0000 LET B=B-24.0000
200 PAUSE DMS B
250 IF DMS B>(C+.0003) AND DMS B>(C-.0003) GOSUB 600
300 GOTO 130
600 BEEP 1
610 GOTO 600

```

Laufzeitermittlung

Beschreibung:

Die meisten Zählwerke in Recordern arbeiten mechanisch und werden entweder von der Auf- oder Abwickelspule angetrieben. Da sich der Durchmesser dieser Wickel ständig verändert, und damit auch die Zählwerksanzeige nicht mit kontinuierlicher Geschwindigkeit abläuft, ist eine Ermittlung von Bandlaufzeiten nicht einfach.

Das Programm 'LAUFZEIT' erleichtert Ihnen diese Arbeit ungemein.

Nach dem Programmstart mit RUN oder DEF"A" erfolgt die Eingabe der Band-Art: (V)ideo / (C)ompact-Cassette.

Dann wird die Eingabe der Gesamtlaufzeit in Minuten erwartet.

 Achtung! Wichtig!

Wenn Sie (V) eingegeben haben, nimmt der PC aus den Programmzeilen 60, 70 und 80 feste Werte für das Zählwerk des Video-Recorders des Autors. Sie können diese Zeilen für Ihren eigenen Recorder ändern, indem Sie die Zählwerks-Anzeige nach genau 10 Minuten (X) und nach 20 Minuten (Y) für die von Ihnen verwendeten Bandarten ermitteln und als X und Y einarbeiten.

Falls weder 'V' noch 'C' bei der Band-Art eingegeben wird, fragt das Programm die Werte für X und Y automatisch ab, so dass das Programm an jeden Recorder und jedes Zählwerk angepasst werden kann!

Als drittes wird entweder eine Tabellen-Erstellung oder die Zeit-Differenz-Berechnung zwischen zwei Zählwerksanzeigen gewählt.

Bei einer Tabelle muss der Anfangs-Zählwerksstand und der gewünschte Tabellen-Intervall eingegeben werden. Es erscheint die Anzeige des Zählerstandes, der abgelaufenen und der restlichen Zeit in Minuten und Sekunden in der Form xxx.xx

Falls ein SHARP Printer angeschlossen ist, werden die Werte auch ausgedruckt.

Zur Ermittlung der Zeitdifferenz zwischen zwei Zählerständen werden diese als ZW1 und ZW2 eingegeben. Es erscheint die Zeit in der gleichen Form.

Sollte die Gesamtlaufzeit überschritten sein, wird dies angezeigt (OVERDRIVE).

Folgende Einsprungmarken sind optional:

DEF"A" - Programmstart (=RUN)
 DEF"N" - Neustart ohne Titel
 DEF"C" - Continue (Tabelle)
 DEF"D" - Differenzen-Berechnung
 DEF"Z" - Neueingabe für Tabellen

Listing:

```

10 "A":PRINT "LAUFZEITEN-BERECHNUNG V.TON-
    UND VIDEO-CASSETTEN" 20 "N"
30 INPUT "ART?(C/V)";A$
40 INPUT "GES.LAUFZEIT(MIN)=";G:IF A$="C"
    THEN LET G=G/2
50 IF A$<"V" THEN GOSUB 800
60 IF A$="V" AND G=120 THEN LET X=202:Y=371.75
70 IF A$="V" AND G=180 THEN LET X=202:Y=372.25
80 IF A$="V" AND G=240 THEN LET X=206:Y=384
90 INPUT "1-TABELLE 2-DIFFERENZ";B$
100 GOTO 200*VAL B$
200 "Z":INPUT "AB ZAEHLER=";Z:USING
210 INPUT "INTERVALL=";D
270 IF A$="C" THEN PRINT "TABELLE FUER C ";G*2
280 IF A$="V" THEN PRINT "TABELLE FUER VHS E ";G
290 CLS
300 "C":GOSUB 1000
310 IF R<0 THEN LET R=G-T
330 CURSOR 0:WAIT 0:USING "####"
  
```

```

340 PRINT "BEI";Z:USING "#####.##":CURSOR 11:
    PRINT "ZEIT=";T
342 CURSOR 35:PRINT "REST=";R
344 LPRINT Z;T;R
350 IF R<0 THEN CURSOR 24:PRINT "OVERDRIVE!":BEEP 1
360 WAIT:PRINT
370 Z=Z+D
390 GOTO 300
400 "D"
410 INPUT "ZW 1=";Z1
420 INPUT "ZW 2=";Z2
470 Z=Z1:GOSUB 1000:Q1=Q
475 Z=Z2:GOSUB 1000:Q2=Q:Q=Q2-Q1
480 H=Q-INT Q:H=H/1.66667:Q=INT Q+H:Q=Q+.005
490 Q=INT(Q*100)/100
500 USING:WAIT 0:PRINT Z1;"->";Z2;"=";Q;" MIN"
510 IF R<0 THEN CURSOR 24:PRINT "OVERDRIVE!":BEEP 1
520 WAIT:PRINT:GOTO 400
800 INPUT "ZAEHLWERK NACH 10 MIN=";X
810 INPUT "ZAEHLWERK NACH 20 MIN=";Y
850 RETURN
1000 REM UPRO
1010 M=(10*Y-20*X)/((X-Y)*X*Y)
1020 N=(20-Y*Y*M)/Y
1030 T=Z*Z*M+Z*N:R=T:Q=T
1040 S=T-INT T:S=S/1.666667
1050 T=INT T+S+.005:T=INT(T*100)/100
1060 R=G-R
1070 L=R-INT R:L=L/1.66667
1080 R=INT R+L+.005:R=INT(R*100)/100
1090 RETURN

```

Kalender

Beschreibung:

Wenn Sie nicht gerade den passenden Kalender zur Hand haben, hilft Ihnen dieses Programm mit Sicherheit weiter.

Nur zwischen den Jahren 1901 bis 2099 rechnet das Programm richtig. Diese Einschränkung gilt deshalb, damit das Programm durch die weiter in der Vergangenheit oder Zukunft liegenden Abweichungen bzw. durch weggefallene Schaltjahre nicht unnötig aufgebläht wurde.

Nach dem Programmstart mit RUN oder DEF"A" erfolgt die Eingabe des Tages, des Monats und des Jahres, wobei jede Eingabe mit (ENTER) abgeschlossen werden muss. Führende Nullen brauchen nicht eingegeben zu werden, jedoch das Jahr immer vollständig. Beispiel: 11.7.1942 als 11(ENTER) 7(ENTER) 1942(ENTER).

Als Ergebnis erscheint nun das eingegebene Datum, dessen Wochentag, die Wochenummer, in welche das Datum fällt, und die Zeichen (+/-). Nun können Sie nämlich das Datum durch Angabe einer beliebigen Anzahl von Tagen (+ oder -) vor bzw. zurückrechnen.

Mit DEF"S" kann ein zweites Datum in der gleichen Form errechnet werden, jedoch ohne die Möglichkeit des Vor- oder Zurückrechnens.

Mit DEF"D" kann dann die Differenz zwischen beiden Daten ausgegeben werden und zwar in Tagen und jeweils nach Tippen von (ENTER) auch in Wochen, Monaten und Jahren.

Mit DEF"Z" und DEF"X" können die beiden Daten jederzeit wieder in die Anzeige geholt werden.

Listing:

```

1  DIM W$(6)
2  DATA "MONTAG", "DIENSTAG", "MITTWOCH", "DONNERSTAG",
   "FREITAG", "SAMSTAG", "SONNTAG"
3  FOR I=0 TO 6:READ W$(I):NEXT I
30 "A":CLS:INPUT "TAG:";T:INPUT "MONAT:";M:
   INPUT "JAHR:";J
40 X=INT((J-1901)*365.25)+T+INT((158*M-157)/5) (M>2)
   *(M+(INT(J/4)<J/4))
50 W1=X-7*INT(X/7)
55 N=X-INT((J-1901)*365.25)
56 WX=INT((N+9-W1)/7)
57 D=0:IF WX<1 THEN LET WX=52:D=-1
58 IF WX>=53 THEN LET WX=1:D=1
70 "Z":CLS:USING:WAIT 0:PRINT T;M;J;"->";W$(W1):
   CURSOR 24:PRINT WX;"WOCHE ";J+D
80 WAIT 0:GOTO 200
130 "S":CLS:INPUT "2.TAG:";T2:INPUT "2.MONAT:";M2:
   INPUT "2.JAHR:";J2
140 Y=INT((J2-1901)*365.25)+T2+INT((158*M2-157)/5)
   -(M2>2)*(M2+(INT(J2/4)<J2/4))
150 W2=Y-7*INT(Y/7)
155 N=Y-INT((J2-1901)*365.25)
156 WY=INT((N+9-W2)/7)
157 D=0:IF WY<1 THEN LET WY=52:D=-1
158 IF WY>=53 THEN LET WY=1:D=1
170 "X":CLS:USING:WAIT 0:PRINT T2;M2;J2;"->";W$(W2):
   CURSOR 24:PRINT WY;"WOCHE ";J2+D:WAIT:PRINT
180 "D":DT=Y-X:DW=DT/7:DM=DT/30.4375:DJ=DT/365.25
185 USING "*****.##":WAIT:CLS
190 PRINT DT;" TAGE":PRINT DW;" WOCHEN":PRINT DM;
   " MONATE":PRINT DJ;" JAHRE":END
200 "C":CURSOR 39:INPUT "(+/-)";Z
220 X=X+Z:CLS
400 J=INT(X/365.25)+1901-(X/365.25=INT(X/365.25))
410 N=X-INT((J-1901)*365.25)
420 FOR O=13 TO 1 STEP-1
430 S=INT((158*O-157)/5)-(O>2)*(O+1-(INT(J/4)=J/4))
440 IF N>S THEN LET T=N-S:M=O:O=0
450 NEXT O:GOTO 50
999 "M":CSAVE "KALENDER"

```

Canasta-Abrechnung

Beschreibung:

Auf besonderen Wunsch des Autors werden hier an dieser Stelle Adresse und Anschrift genannt:

Hendrik Ambrosius Cube-Soft Goerdelerstr. 29 2400 Lübeck 1

Mit diesem Programm macht selbst das Abrechnen der Punkte beim Canasta-spielen Spaß. Es berechnet von selbst den Gesamtpunktstand und sagt, ob jemand schon die 5000-Punkte-Grenze überschritten hat und mit wieviel man 'raus' muß. Außerdem wird angezeigt, an welcher Stelle die einzelnen Spieler stehen.

Wenn man ein neues Spiel beginnt, muß man die Namen der Spieler eingeben. Nachher wird dann der Name des Spielers angezeigt, für den die Punkte eingegeben werden sollen. Man muß alle Textausgaben mit ENTER bestätigen. Um Strom zu sparen kann man während des Spieles den Rechner ausschalten. Man sollte in dem Moment, wo der Rechner "PUNKTE EINGEBEN" anzeigt, den Schalter betätigen und nach der Runde das Programm mit DEF+A wieder starten. Vorsicht! Kein RUN oder DEF+D eingeben, wenn das Spiel noch nicht beendet ist, sonst werden alle Daten gelöscht.

Nach der Eingabe der Punkte wird eine Statusanzeige für jeden einzelnen Spieler erstellt, die folgendermaßen aufgebaut ist:

Punktstand, Raus (mit wievielen Punkten man 'raus' muß), Platzierung Und hier noch einmal die Tastenkommandos des Programms im Überblick:

- RUN - Neues Spiel mit neuen Spielern starten.
- DEF+D - Neues Spiel mit den gleichen Spielern wie beim vorigen Spiel.
- DEF+S - Statusanzeige.
- DEF+A - Punkte einer neuen Runde eingeben.
- DEF+H - Kleine Hilfstexte zur DEF-Tastenbelegung.

Listing:

```

2 FOR I=1 TO 10 STEP .5
3 WAIT I
5 PRINT "*CANASTARECHNER*":NEXT I
6 PAUSE "*CANASTARECHNER*":WAIT
7 PRINT "(C) 86 CUBE-SOFT":PRINT "AND H.AMBROSIUS"
10 INPUT "ANZ.SPIELER:";Z
20 DIM P$(Z)*16,S(Z),J$(0)*16
21 FOR I=1 TO Z
22 PRINT "NAME V. SP. ";I:INPUT ">";P$(I)
23 NEXT I
25 "D"G=0:FOR I=1 TO Z:S(I)=0:NEXT I:E=0
30 "A"
40 PRINT "PUNKTE EINGEBEN":G=G+1
45 PRINT G;"RUNDE"
50 FOR I=1 TO Z
60 PRINT P$(I):INPUT ">";C:S(I)=S(I)+C
70 NEXT I
80 "S" FOR I=1 TO Z
82 PRINT "STATUSANZEIGE F.":PRINT P$(I)
85 J$(0)=" "
90 J$(0)=J$(0)+STR$(S(I))

```

```

200 IF S(I)<0 THEN LET J$(0)=J$(0)+" 30"
210 IF S(I)>=1500 AND S(I)<3000 THEN LET J$(0)=J$(0)+" 90"
220 IF S(I)>=3000 THEN LET J$(0)=J$(0)+" 120"
225 IF S(I)>=0 AND S(I)<1500 THEN LET J$(0)=J$(0)+" 50"
230 O=0:FOR K=1 TO Z
240 IF S(K)>S(I)THEN LET O=O+1
250 NEXT K:O=O+1
255 IF O=1 AND S(I)>=5000 THEN LET E=I
260 J$(0)=J$(0)+" "+STR$(O)
270 PRINT J$(0):NEXT I
275 IF E<>0 THEN PRINT P$(E):PRINT "HAT GEWONNEN !":
CALL 1444:BEEP 2:PRINT "ICH GRATULIERE !":END
279 PRINT "NEUES SPIEL,":PRINT " NEUES GLUECK !"
280 GOTO "A"
300 "H"
310 PRINT "DEF-H HILFE"
320 PRINT "DEF-A NEUE PUN-":PRINT "KTE EINGEBEN"
330 PRINT "DEF-S STATUSANZ.":PRINT "ANSEHEN"
350 PRINT "RUN NEUES SPIEL"
360 PRINT "DEF-D NEUES":PRINT "SPIEL MIT DEN":
PRINT "GLEICHEN":PRINT "SPIELERN"

```

Ilford-Verfahren

Beschreibung:

Programm das bei der Entwicklung von Schwarz-Weiß-Filmen nach dem Ilford-Verfahren hilft.

Das Programm ist für Fotofans gedacht die gerne ihr Schwarz-Weiß-Filme selbst entwickeln. Das Programm bezieht sich auf die Entwicklungsmethode die von ILFORD für deren SW-Filme empfohlen wird. Das heißt: pro Entwicklungsminute 10 sek. Dose kippen, 50 sek. Ruhe. Das Programm läßt sich natürlich sehr einfach an andere Entwicklungsabläufe anpassen. Vorteil gegenüber der sonst üblichen Stoppuhr ist, daß man nicht dauernd auf die Uhr schauen muß, sondern sich auf den Beeper verlassen kann.

Die Bedienung des Programmes ist sehr einfach. Man gibt die Gesamtentwicklungsdauer in vollen Minuten ein und startet daß Programm sobald der Entwickler vollständig in der Dose ist mit der Enter-Taste. Das Display zeigt die Anweisungen an und der Beeper zeigt die Veränderung des Anweisung an. Beendet ist der Entwicklungsablauf, wenn der Beeper fünfmal ertönt.

Listing:

```

5 "A"
10 PAUSE "TIMER"
20 INPUT "ENTW.ZEIT T=";T
30 FOR N=1 TO T
40 WAIT 544
50 PRINT "KIPPEN"
60 BEEP 1
70 WAIT 2900
80 A$="ENTW."
90 S$="MIN"
100 D$="T="

```



```
110 PRINT A$;D$;N;S$
120 BEEP 2
130 NEXT N
140 BEEP 5
150 WAIT 100
160 PRINT "FERTIG"
170 GOTO 20
180 END
```

Ilford-Cibachrome-Verfahren

Beschreibung:

Programm das bei der Entwicklung von Schwarz-Weiß-Filmen nach dem Ilford Cibachrome Prozeß hilft.

Das Programm ist der entsprechende Prozesstimer für den Ilford Cibachrome Prozeß P30 bei 24 Grad Celsius.

Listing:

```
10 "C"
20 PAUSE"CIBACHROME P30"
30 PAUSE"T=24 GRD.C"
40 PRINT"START ENTW.3MIN"
50 FOR N=1 TO 2
60 BEEP N
70 WAIT 3470
80 S$="ENTW."
90 D$="MIN."
100 PRINT S$;N;D$
110 NEXT N
120 BEEP3
130 FOR I=0 TO 60
140 WAIT 52
150 Z$="NOCH "
160 X$=" SEK."
170 PRINT Z$;60-I;X$
180 NEXT I
190 WAIT
200 BEEP6
210 PAUSE"WASSERN 0.5MIN"
220 FOR D=0 TO 30
230 WAIT 52
240 PRINT Z$;30-D;X$
250 NEXT D
260 WAIT1735
270 PAUSE"BLEICH 3MIN"
280 FOR N=1TO2
290 BEEP N
300 WAIT 3470
310 F$="BLEICH"
320 PRINT F$;N;D$
330 NEXT N
340 BEEP 3
350 FOR I=0 TO 60
```

```
360 WAIT 52
370 PRINT Z$;60-I;X$
380 NEXT I
390 BEEP 6
400 PAUSE "FIXIER 3MIN"
410 FOR N=1 TO 2
420 BEEPN
430 WAIT3470
440 H$="FIXIER"
450 PRINT H$;N;D$
460 NEXT N
470 BEEP 3
480 FOR I=0 TO 60
490 WAIT 52
500 PRINT Z$;60-I;X$
510 NEXT I
520 BEEP 6
530 PRINT "TROMMEL LEEREN"
540 WAIT 200
550 PRINT "FERTIG"
560 WAIT
570 GOTO 20
580 END
```

Tental-UK 3 Prozeß

Beschreibung:

Programm das bei der Entwicklung von Schwarz-Weiß-Filmen nach dem Tetenal UK-3 Prozess oder durch Anwendung des Color-Umkehr-Kit hilft.

Das Programm ist ebenfalls für Fotolaboranten gedacht, die den Tetenal Prozess UK-3 oder das Color-Umkehr-Kit verwenden. Das Programm ist für den Prozess bei 35 Grad geschrieben und funktioniert ähnlich wie das Programm "Ilford-Verfahren". Alle Zustandsänderungen werden im Display angezeigt und durch BEEPton signalisiert.

Listing:

```
1500 ", "
1510 PAUSE "TETENAL"
1520 PAUSE "T=35 GRAD"
1530 PRINT "START VORW.1MIN"
1540 WAIT 2690
1550 PRINT "1MIN. VORWAERM."
1570 FOR I=1 TO 10
1580 BEEP 1
1590 WAIT 52
1600 Z$="NOCH"
1610 X$="SEK. "
1620 PRINT Z$;10-I;X$
1630 NEXT I
1640 BEEP 3
1650 WAIT 520
1655 E$="EINFUEL"
1656 L$="LEN"
```

```
1660 PRINT ES;L$
1665 FOR I=1 TO 105
1670 WAIT 50
1675 IF I>90 THEN BEEP 1
1680 PRINT Z$;105-I;X$
1690 NEXT I
1700 WAIT 390
1710 PRINT ES;L$
1720 FOR I=1 TO 4
1725 FOR N=0 TO 29
1730 WAIT 52
1740 PRINT Z$;N-30;X$
1750 NEXT N
1755 BEEP I
1760 NEXT I
1770 WAIT 390
1780 PRINT ES;L$
1790 FOR I=1 TO 4
1800 WAIT 3470
1805 BEEP 1
1810 PRINT "FARBE";I;"MIN."
1820 NEXT I
1830 FOR N=0 TO 29
1840 WAIT 52
1850 PRINT Z$;30-N;X$
1860 NEXT N
1870 WAIT 390
1880 PRINT ES;L$
1890 FOR I=0 TO 29
1900 WAIT 52
1910 IF I>19 BEEP 1
1920 PRINT Z$;30-I;X$
1930 NEXT I
1940 WAIT 390
1950 PRINT ES;L$
1960 WAIT 3470
1970 PRINT "BLEICH 1.MIN."
1980 FOR I=0 TO 59
1990 WAIT 52
2000 IF I>49 THEN BEEP 1
2010 PRINT Z$;60-I;X$
2020 NEXT I
2030 BEEP 6
2040 WAIT
2050 PAUSE "BITTE WAESSERN"
2060 GOTO 1530
```

Einheitenumrechnung

Beschreibung:

Programm zum umrechnen gängiger Einheiten in weniger übliche Einheiten. Da oftmals Umrechnungsparameter nicht unmittelbar greifbar sind, ist es manchmal sehr langwierig den Wert in der gewünschten Einheit zu erhalten.

Einheitenumrechnung:

Start	: DEF	"A"	Zentimeter	->	Zoll
		"Z"	Zoll	->	Zentimeter
		"S"	Meter	->	Fuß
		"X"	Fuß	->	Meter
		"D"	Kilometer	->	Meilen
		"C"	Meilen	->	Kilometer
		"F"	Kilogramm	->	Pound
		"V"	Pound	->	Kilogramm
		"G"	Gramm	->	Unzen
		"B"	Unzen	->	Gramm
		"H"	Liter	->	Quarts
		"N"	Quarts	->	Liter
		"J"	Celsius	->	Fahrenheit
		"M"	Fahrenheit	->	Celsius
		"K"	m/s	->	km/h
		"SPACE"	km/h	->	m/s

Listing:

```

5 CLEAR:END
10 "A" PRINT "ZENTIMETER>ZOLL"
20 INPUT "CM:";A:B=A*.39
30 PRINT "ZOLL:";B:END
40 "Z" PRINT "ZOLL>ZENTIMETER"
50 INPUT "ZOLL:";A:B=A*2.54
60 PRINT "CM:";B:END
70 "S" PRINT "METER>FUSS"
80 INPUT "METER:";A:B=A*3.28
90 PRINT "FUSS:";B:END
100 "X" PRINT "FUSS>METER"
110 INPUT "FUSS:";A:B=A*.30487
120 PRINT "METER:";B:END
130 "D" PRINT "KM > MEILEN"
140 INPUT "KM:";A:B=A*.621
150 PRINT "MEILEN:";B:END
160 "C" PRINT "MEILEN > KM"
170 INPUT "MEILEN:";A:B=A*1.609
180 PRINT "KM:";B:END
190 "F" PRINT "KG > POUND"
200 INPUT "KG:";A:B=A*2.2
210 PRINT "POUND:";B
220 "V" PRINT "POUND > KG"
230 INPUT "POUND:";A:B=A*.45
240 PRINT "KG:";B:END
250 "G" PRINT "GRAMM > UNZEN"
260 INPUT "GRAMM:";A:B=A*.035
270 PRINT "UNZEN:";B:END
280 "B" PRINT "UNZEN > GRAMM"
290 INPUT "UNZEN:";A:B=A*28.35
300 PRINT "GRAMM:";B:END
310 "H" PRINT "LITER > QUARTS"
320 INPUT "LITER:";A:B=A*1.0567
330 PRINT "QUARTS:";B:END
340 "N" PRINT "QUARTS > LITER"
350 INPUT "QUARTS:";A:B=A*.946
360 PRINT "LITER:";B

```

```

370 "J" PRINT "GRAD C > GRAD F"
380 INPUT "GRAD C: ";A:B=9*A/5+32
390 PRINT "GRAD F: ";B:END
400 "M" PRINT "GRAD F > GRAD C"
410 INPUT "GRAD F: ";A:B=5/9*(A-32)
420 PRINT "GRAD C: ";B:END
430 "K" PRINT "M/S > KM/H"
440 INPUT "M/S: ";A:B=A*3.6
450 PRINT "KM/H: ";B:END
460 " " PRINT "KM/H > M/S"
470 INPUT "KM/H: ";A:B=A*.277
480 PRINT "M/S: ";B:END

```

Chemische Formeln

Beschreibung:

Dieses Programm ist für alle diejenigen gedacht, die sich etwas schwer tun mit chemischen Formeln. Nach Eingabe einer chemischen Formel wird umgekehrt der Name dieser Formel verbal angezeigt und umgekehrt.

Beispiel: Run mit "DEF N"

Eingabe chemische Formel/Name der Formel

Bemerkung: Das Programm kann je nach Speicherplatz ausgebaut werden. Bei den Namen der Formeln wurde bei den Elementen das chemische Symbol eingegeben.

Listing:

```

3100 END
3110 PRINT A$(0,N):END
3120 PRINT A$(1,N):END
3200 "N":CLEAR:PAUSE " -----CHEM.FORMELN----- "
3210 PRINT " FO>NA(F);NA>FO(N)"
3220 I$=INKEY$:B=ASC I$
3230 IF B=70 THEN 3260
3240 IF B=78 THEN 3310
3250 GOTO 3220
3260 PAUSE "WELCHE FORMEL?":INPUT FOS
3270 DIM A$(1,100):GOSUB 3400
3280 FOR N=0 TO 100
3290 IF A$(1,N)=FOSGOTO 3110
3300 NEXT N
3305 PRINT " NAME NICHT ENTHALTEN !":END
3310 PAUSE "WELCHER NAME?":INPUT NAS
3320 DIM A$(1,100):GOSUB 3400
3330 FOR N=0 TO 100
3340 IF A$(0,N)=NASGOTO 3120
3350 NEXT N
3360 PRINT "FORMEL NICHT ENTHALTEN !":END
3400 A$(0,1)="CL.SAEURE"
3410 A$(1,1)="HCLO3"
3420 A$(0,2)="BUTAN"

```

3430 A\$(1,2) = "C4H10"
3440 A\$(0,3) = "PERCL.SAEURE"
3450 A\$(1,3) = "HCLO4"
3460 A\$(0,4) = "P.SAEURE"
3470 A\$(1,4) = "H3PO4"
3480 A\$(0,5) = "SALPETERSAEURE"
3490 A\$(1,5) = "HNO3"
3500 A\$(0,6) = "AMEISENSAEURE"
3510 A\$(1,6) = "H2CO2"
3520 A\$(0,7) = "OXALSAEURE"
3530 A\$(1,7) = "H2C2O4"
3540 A\$(0,8) = "NATRONLAUGE"
3550 A\$(1,8) = "NAOH"
3560 A\$(0,9) = "KALILAUGE"
3570 A\$(1,9) = "KOH"
3580 A\$(0,10) = "K.PHOSPHAT"
3590 A\$(1,10) = "K3PO4"
3600 A\$(0,11) = "SODA"
3610 A\$(1,11) = "NA2CO3"
3620 A\$(0,12) = "PYRIDIN"
3630 A\$(1,12) = "H5C5N"
3640 A\$(0,13) = "NA.KARBONAT"
3650 A\$(1,13) = "NA2CO3"
3660 A\$(0,14) = "CL.OFORM"
3670 A\$(1,14) = "CHCL3"
3680 A\$(0,15) = "NA.DIKARBONAT"
3690 A\$(1,15) = "NAHCO3"
3700 A\$(0,16) = "NA.NITRAT"
3710 A\$(1,16) = "NANO3"
3720 A\$(0,17) = "NA.SULFAT"
3730 A\$(1,17) = "NA2SO4"
3740 A\$(0,18) = "KALK"
3750 A\$(1,18) = "CACO3"
3760 A\$(0,19) = "GIPS"
3770 A\$(1,19) = "CASO4"
3780 A\$(0,20) = "P.OXID"
3790 A\$(1,20) = "P4O10"
3800 A\$(0,21) = "HYDRAZIN"
3810 A\$(1,21) = "N2H4"
3820 A\$(0,22) = "PHOSGEN"
3830 A\$(1,22) = "COCL2"
3840 A\$(0,23) = "ACETYLEN"
3850 A\$(1,23) = "C2H2"
3860 A\$(0,24) = "FORMALDEHYD"
3870 A\$(1,24) = "CH2O"
3880 A\$(0,25) = "ESSIGSAEURE"
3890 A\$(1,25) = "C2H4O2"
3900 A\$(0,26) = "OKTAN"
3910 A\$(1,26) = "C8H18"
3920 A\$(0,27) = "ROHRZUCKER"
3930 A\$(1,27) = "C12H22O11"
3940 A\$(0,28) = "CA.PHOSPHAT"
3950 A\$(1,28) = "CA3(PO4)2"
3960 A\$(0,29) = "K.PHOSPHAT"
3970 A\$(1,29) = "KH2PO4"
3980 A\$(0,30) = "BLAUSAEURE"
3990 A\$(1,30) = "HCN"
4000 A\$(0,31) = "PHOSPIN"
4010 A\$(1,31) = "PH3"

4020 A\$(0,32)="LI.HYDRID"
4030 A\$(1,32)="LIH"
4040 A\$(0,33)="AG.NITRAT"
4050 A\$(1,33)="AGNO3"
4060 A\$(0,34)="DIPHOSPHIN"
4070 A\$(1,34)="P2H4"
4080 A\$(0,35)="PYRIT"
4090 A\$(1,35)="FES2"
4100 A\$(0,36)="PROPAN"
4110 A\$(1,36)="C3H8"
4120 A\$(0,37)="NA.THIOSULFAT"
4130 A\$(1,37)="NA2S2O3"
4140 A\$(0,38)="AETHAN"
4150 A\$(1,38)="C2H6"
4160 A\$(0,39)="AMINOMETHAN"
4170 A\$(1,39)="CH5N"
4180 A\$(0,40)="ALKOHOL"
4190 A\$(1,40)="C2H6O"
4200 A\$(0,41)="NICOTIN"
4210 A\$(1,41)="C10H14N2"
4220 A\$(0,42)="KRYOLITH"
4230 A\$(1,42)="NA3(ALF6)"
4240 A\$(0,43)="CAMPHER"
4250 A\$(1,43)="C100H16"
4260 A\$(0,44)="KOHLENSAEURE"
4270 A\$(1,44)="H2CO3"
4280 A\$(0,45)="S.SAEURE"
4290 A\$(1,45)="H2SO4"
4300 A\$(0,46)="BENZOLSAEURE"
4310 A\$(1,46)="C7H6O2"
4320 A\$(0,47)="ZN.BLENDE"
4330 A\$(1,47)="ZNS"
4340 A\$(0,48)="ETHANOL"
4350 A\$(1,48)="C2H6O"
4360 A\$(0,49)="H.PEROXID"
4370 A\$(1,49)="H2O2"
4380 A\$(0,50)="K.PERMANGANAT"
4390 A\$(1,50)="KMNO4"
4400 A\$(0,51)="NA.PERCL.AT"
4410 A\$(1,51)="NACLO4"
4420 A\$(0,52)="NA.CL.AT"
4430 A\$(1,52)="NACLO3"
4440 A\$(0,53)="AMMONIUMDICR.AT"
4450 A\$(1,53)="(NH4)2CR2O7"
4460 A\$(0,54)="ACETALDEHYD"
4470 A\$(1,54)="C2H4O"
4480 A\$(0,55)="K.BICR.AT"
4490 A\$(1,55)="K2CR2O7"
4500 A\$(0,56)="NA.AMID"
4510 A\$(1,56)="NA2NH2"
4520 A\$(0,57)="K.SULFAT"
4530 A\$(1,57)="K2SO4"
4540 A\$(0,58)="CR.TRISULFAT"
4550 A\$(1,58)="CR2(SO4)3"
4560 A\$(0,59)=" "
4570 A\$(1,59)=" "
6000 RETURN

Gehäuseberechnung von Lautsprechern

Beschreibung:

Hilfsprogramm zur Gehäuseberechnung von Lautsprecherboxen.

Es werden Vorschläge zur Gehäuseart gemacht und für verschiedene Volumina die sich ergebenden Grenzfrequenzen sowie Gütefaktoren (wichtig für Einschwingverhalten) berechnet bzw. berücksichtigt und andere Parameter bestimmt.

Bei unbekanntem Daten der fraglichen Chassis, und zur Überprüfung der Herstellerangaben kann man mit Hilfe des Programms und eines Testaufbaus alle wichtigen Parameter ermitteln:

Benötigt wird hierzu ein durchstimmbarer Tongenerator mit Leistungsausgang (ggfs. Hi-Fi - Endstufe) nicht die Leistung ist hier wichtig, sondern der Spannungshub, ca. 11Volt eff. werden benötigt.

4 Fünfmarkstücke (zus. 40 gr.), die als Hilfsmasse dienen und zu diesem Zweck vorsichtig auf die Membran aufgebracht werden - Tesafilm -.

Zum Programm:

Zuerst werden die Chassisdaten erfragt, welche man den Herstellerunterlagen entnimmt. Sind diese nicht alle vorhanden oder zweifelhaft (soll sogar bei den "Edelmarken" vorkommen), so bedient man sich der zweiten Abfrage und bestimmt die Daten selbst (hierbei erlebt man die größten Überraschungen - s.o.). Bei Unklarheiten bitte ins Listing schauen! reichlich REM's.

Aus den Daten ermittelt das Programm alle weiteren zur Gehäuseberechnung notwendigen Parameter und macht einen Gehäusevorschlag. Man hat dann die Wahl zwischen 4 verschiedenen Formen und steigt in den eigentlichen Rechenvorgang ein.

Geschlossene Gehäuse:

Es erfolgt die Ausgabe eines Listings, in dem die verschiedenen Volumina zu den sich ergebenden Resonanzfrequenzen und den jeweiligen Qts - Werten (Einschwingverhalten - 0.5 und 0.7 sind unterstrichen) ausgeworfen werden.

Bassreflex:

Hier werden weitere Parameter erfragt und im folgenden Listing die Dimensionen des Reflextunnels angegeben. (siehe Hifi-Boxen- selbstgemacht, elrad extra 2)

Auch Hornlautsprecher und Transmission-Lines können ansatzweise berechnet werden.

Listing:

```

2 CLS:PAUSE "LAUTSPRECHERGEHAUSE"
4 INPUT "CE-126P oder CE-140P ? 1 oder 2 >!";DR
5 IF DR<1 OR DR>2 GOTO 4
6 IF DR=1 GOTO 10

```



```

8 OPEN
10 PRINT=LPRINT:CLS:CLEAR:CURSOR 24:WAIT 100:PRINT "
Lautsprechergehaeuse":CURSOR 48:
PRINT " Berechnung !"
11 DIM CBS(0)*80
12 CLS:PRINT "Cassis-Bezeichnung :":REM Wie heisst
der Probant?
14 INPUT CBS(0)
16 PRINT CBS(0)
20 CLS:WAIT 0:PRINT=PRINT:
PRINT "Cassisdaten-Erfassung :":
30 LINE(6,8)-(149,8):REM Abfragen bis Z.160 fuer Herst.
u Messt.Erm.gleich.
40 CURSOR(48):PRINT "Herstellerdat.-Eing. =1"
50 INPUT "Messtechn.Ermittlung =2";DE
60 INPUT "Freiluftresonanz (Hz)=?";FR:REM Chassis
moeglichst frei aufstellen!
65 PRINT=LPRINT:PRINT "Freiluftresonanz =";FR;" Hz":
REM Messaufbau od.Hstd.
70 INPUT "Eff.Membr.Durchm.(cm)=?";D:REM Zur Not
messen - halbe Sicke!
72 IF D<>0 LET D=INT(D*10+.5)/10:GOTO 78
73 REM Wird 0 eingeg.dann Flaeche ang.
74 INPUT "Eff.Membr.Fl.(qcm)";F
76 D=2*SQR(F/PI):D=INT(D*10+.5)/10
78 PRINT "Eff.Membr.Durchm.=";D;"cm":
REM Kann man auch ber.
80 INPUT "Spulenwiderstand(Ohm)=?";RE:
REM Ein Ohmmeter wird wohl zur Hand sein?
85 PRINT "Spulenwiderstand =";RE;"Ohm"
90 IF DE=2 THEN 160
95 REM Jetzt gehts gleich los
100 INPUT "Eff.Membranmasse (gr)=?";MM:
REM wer kennt die schon?
105 PRINT "Eff.Membr.Masse =";MM;"gr"
110 INPUT "Compliance*10E-4 m/N = ?";CL:
REM Nachgiebigkeit der Einspann.
112 C=1E3/(4*PI*PI*FR*FR*MM):USING "###.##^":
REM Kann man auch berechnen!
114 PRINT "Compl. HA =";CL;"m/N":REM Na, wer hat recht?
116 PRINT "Compl. Re =";C;"m/N":USING:REM Bitteschoen!
120 INPUT "Freiluftguete Qts =?";QT:
REM Kennen wir die denn?
125 PRINT "Freiluftguete Qts=";QT:
REM Wir werden's schon herausfinden.
130 ME=MM/1E3
150 GOTO 460
160 INPUT "Gew.der Hilfsmass(gr)=?";MH:
REM 4x5DM-->40gr.
170 INPUT "Res.m.Hilfsmasse (Hz)=?";FH:
REM Versuchsaufbau mit genauem 10 Ohm
180 MM=MH/(((FR*FR)/(FH*FH))-1):
REM Widerstand Eichen, je nach Instrument im
190 ME=MM/1E3:REM 100 mV od.1V Bereich auf 1/10 der
Skala mit der Ausgangssp.
200 C=1/(4*PI*PI*FR*FR*ME):REM Einstellen und dann
nicht mehr veraendern.
230 INPUT "Imped.bei Reson.(Ohm)=?";RM:
REM Jetzt kann direkt in 'Ohm' abgel.

```

```

240 VA=.863*(D^4)*C:REM werden
250 R0=RM/RE:REM Hier wird der Gleichstromwiderst.gebr.
    (siehe oben)
260 R1=RE*SQR(R0):REM und zu Kontrollzwecken
    herangezogen.
265 WAIT 0:PRINT=PRINT
270 PRINT "Untere Frequenz fuer";R1:
    REM Hoffendlich hat der Genertator eine
275 INPUT "Ohm Impedanz    (Hz)=?";F1:
    REM genaue Skala oder Sie einen Frequenz-
290 PRINT "Obere Frequenz fuer";R1:
    REM zaehler !?!Sie veraendern also solange
295 INPUT "Ohm Impedanz    (Hz)=?";F2:
    REM die Frequenz, bis der Impedanzwert
310 QM=(FR*(SQR(R0)))/(F2-F1):
    REM auf dem Voltmeter angezeigt wird.
320 QE=QM/(R0-1):REM Ist doch garnicht so schwer, oder ?
    !? So, wenn jetzt alles
330 QT=(QM*QE)/(QM+QE):REM richtig war, kennen wir
    jetzt die 'echte' Compl.
400 CLS:PRINT=LPRINT:PRINT "Compl. C=";C:
    REM Kennt der Hersteller sie auch ?
410 PRINT "Membr.M.=";MM;"gr":
    REM Und das alles ohne Schere
420 WAIT 100:PRINT "Frei Qts=";QT:
    REM Ein sehr wichtiger Wert.
460 TA=FR/QT:WAIT 100:PRINT=LPRINT:
    REM Faustformel zur ueberschlaegigen Einordng.
470 IF TA<=40 THEN CLS:PRINT "    Chassis fuer":
    PRINT "    TRANSMISSIONLINE"
480 IF TA>=30 AND TA<=100 THEN CLS:
    PRINT "    Chassis fuer":
    PRINT "    GESCHLOSSENE BOXEN"
490 IF TA>=60 AND TA<=140 THEN CLS:
    PRINT "    Chassis fuer":PRINT "    BASSREFLEXBOXEN"
500 IF TA>=120 THEN CLS:PRINT "    Chassis fuer":
    PRINT "    HORN-LAUTSPRECHER"
600 CLS:WAIT 0:PRINT=PRINT:PRINT "GESCHLOSSENE BOX = 1"
605 PRINT "BASSREFLEX - BOX = 2"
610 PRINT "HORNLAUTSPRECHER = 3"
615 INPUT "TRANSMISSIONLINE = 4";A
620 ON A GOTO 960,2000,3000,4000
960 CLS:PRINT=LPRINT:PRINT "    GESCHLOSSENE BOX":
    INPUT "LISTENAUSDRUCK >= 1";LA
1000 FG=FR:REM Es werden die zu erwartenden Einbaugueten
    berechnet und gezeigt.
1020 CLS:PRINT CHR$ &7C;"EINBAU-";CHR$ &7C;"NETTO- ";
    CHR$ &7C;"EINBAU-"
1030 PRINT CHR$ &7C;"RESON. ";CHR$ &7C;"VOLUM. ";
    CHR$ &7C;"GUETE"
1040 PRINT CHR$ &7C;" (Hz) ";CHR$ &7C;"(LTR) ";
    CHR$ &7C;" Q ts"
1060 FG=FG+5
1070 QE=(INT(100*FG*QT/FR))/100
1080 A=2.1857E-2*(D^4)
1090 B=ME*((1.15*FG*FG)-(FR*FR))
1100 VN=INT((A/B)+.5)
1130 CURSOR(72):PRINT=PRINT:PRINT "" :WAIT 120:
    PRINT=LPRINT:CURSOR(72):
    PRINT " ";FG;" ";VN;" ";QE

```

```

1140 IF QE<.72 AND QE>.68 THEN 1300
1142 REM Guenstige Werte werden unterstrichen.
1145 IF QE<.52 AND QE>.48 THEN 1300
1147 REM Etwas 'schlanker' aber praezieser Bass.
1150 NZ=FG/15
1160 IZ=INT(NZ)
1170 IF NZ=IZ THEN 1200
1180 GOTO 1060
1200 CLS:INPUT "Weitere Werte J/N ?";WS:
    REM Damit noch etwas Papier ueberbleibt.
1205 IF WS="J" THEN PRINT=PRINT:GOTO 1020
1210 END
1300 LINE(6,31)-(149,31)
1310 GOTO 1060
2000 CLS:PRINT=LPRINT:WAIT 0:PRINT "  BASSREFLEX - BOX":
    REM Tabelle in elrad
2005 INPUT "Res.FB im Geh.(Hz) =";FB:REM Boxenbauheft:
    Passt nicht in den Speicher
2007 PRINT "Reson.FB im Geh. =";FB;"Hz":
    REM des 1350, kann aber bei mir gegen
2010 INPUT "Tunnelartfaktor K =";KT:
    REM Rueckporto bezogen werden.
2015 PRINT "Tunnelartfaktor K=";KT:
    REM In dieser Version nur mit obiger
2020 INPUT "Nettovolumen (ltr) =";NV:
    REM Tabelle zu berechnen.
2025 PRINT "Nettovolumen   =";NV;"lr":
    REM Vielleicht kommt auch ein
2030 DR=4:REM Update
2050 CLS:WAIT 0:PRINT "Reflexkanal"
2055 PRINT "Durchm. Flaeche Laenge"
2060 PRINT " (cm) (qcm) (mm)"
2090 DR=DR+1
2110 RF=INT(DR*DR*PI/4)
2120 RL=INT(((235000*DR*DR)/(FB*FB*NV))-(5*KT*DR))
2126 CURSOR 72:IF RL<0 THEN WAIT:PRINT=PRINT:PRINT DR;
    "Durchm.>o.Vol.<":CURSOR 72:WAIT 0:PRINT ""
2128 IF RL<0 THEN GOTO 2140
2130 CURSOR 72:PRINT=LPRINT:WAIT 100:PRINT DR;" ";
    RF;" ";RL
2140 IF DR=18 THEN 2160
2150 GOTO 2090
2160 INPUT "Weitere Werte J/N ?";WS
2170 IF WS="J" GOTO 2000
2180 END
3000 CLS:PRINT=LPRINT:PRINT "Hornposition ? :":
    REM Ist wohl mehr akademisch,
3015 INPUT "Frei=2 :Wand=4 :Ecke=8";P:
    REM Theorie und Praxis, Sie wissen schon
3016 IF P=2 PRINT "Frei !":GOTO 3020
3017 IF P=4 PRINT "Wand":GOTO 3020
3018 IF P=8 PRINT "Ecke":GOTO 3020
3019 GOTO 3015
3020 INPUT "Unt.Horn-Grenzfr.=";FH
3025 PRINT "Unt.Horn-Grenzfr.=";FH
3030 INPUT "Gescw.-Transf.AT/AH=";W
3035 L=INT(8500/FH)
3038 AT=(D/100)*(D/100)*PI/4
3040 AH=AT/W
3050 T=(42.7*W)/(MM*FR)

```

```
3060 MU=9200/(P*FH*FH)
3070 CLS:PRINT "Mundfl.dZ=6dB=";MU;"qm"
3075 PRINT "Lambda/4-Hornl.=";L;"cm"
3080 PRINT "Horntypkonstante=";T
3090 PRINT " T=0 ";CHR$ &7C;" T=0-1 ";CHR$ &7C;" T=1"
3100 PRINT " Hyper ";CHR$ &7C;"Hyp.-Ex";CHR$ &7C;"Expon."
3110 WAIT:PRINT " Optimal: T=0.7"
3120 CLS:INPUT "Neu=0 OK=1";A
3125 IF A=0 THEN 3000
3130 CLS:WAIT 0:PRINT "Hornquerschnittsform : "
3135 INPUT "rund=1 quadr=2 dreie=3";QF
3140 IF QF=1 THEN 3150
3145 INPUT "Konst.Breite(M)=";KB
3150 INPUT "Entfernung vom Hornhals=";X
3165 M=(4*FH)/343
3170 B=EXP(M*X)
3175 C=EXP(-(M*X))
3180 DH=(B+C)/2
3190 E=T*(B-C)/2
3210 H=(DH+E)^2
3220 AX=AH*H*H
3230 DX=SQR(4*AX/PI)
3240 IF QF=1 THEN PRINT "Durchm=";DX;"m"
3245 IF QF=1 THEN 3150
3250 HB=AX/KB
3260 IF QF=2 THEN PRINT "Breite=";HB;"m"
3270 HB=2*HB
3280 IF QF=3 THEN PRINT "Breite=";HB;"m"
3290 GOTO 3150
4000 TL=340/(FR*2.8)
4010 CLS:PRINT:PRINT "Transmissionline-Laenge=":
    REM Wer haette das gedacht?
4020 PRINT " ";TL;"Meter":
    REM Aber im Ernst, ab jetzt ist der SCHREINER in
4025 END
4030 REM Ihnen gefordert, viel Spass !!!
4040 REM Mein Tip, bauen Sie geschlossene Boxen,
    die Ergebnisse sind bei
4050 REM richtiger Konstruktion (Richtiges Chassis in
    geeigneter Box) immer
4060 REM noch am 'Besten', aber hoeren Sie selbst !!!
```

Berechnung von aktiven Filtern

Beschreibung:

Hilfsprogramm zur Berechnung von aktiven Filtern.

Das Programm dient zur Berechnung von aktiven Filtern 1.- 10. (40.) Ordnung bei unterschiedlichster Charakteristik und ist selbsterklärend. Das jeweilige Schaltungsdiagramm wird als kleine Grafik auf dem Display des 1350 mit ausgegeben.

Auch zur Filterwirkung bezügl. Frequenzgang kann eine Grafik aufgerufen werden, dauert aber lange.

Listing:

```

10 "I"CLS:WAIT0:PRINT"DEF A => Filter 1.-40.O.":
    PRINT"DEF B => Filter 1.u.2.O."
15 WAIT:PRINT"DEF F => Filterwirkung (auf Wunsch mit
    Grafik!)"
20 END
30 "B":CLS:CLEAR:X=6:WAIT 0:PRINT "Dimensionierung
    aktiver Filter 1. und 2. Ordnung":PRINT ""
32 OOS="4C52610161524C"
35 INPUT "Filterordnung:<1 od.2>?";N
40 IF(N-1)*(N-2)<>0 THEN 30
50 IF N=2 THEN 290
60 CLS:PRINT "Filter 1. Ordnung"
70 PRINT "Werte f.R,C u.fg:2<>0 u.den unb.=0 eingeben!"
80 INPUT "Widerstand R in kOhm,?";R
100 IF R<0 THEN 80
110 INPUT "Kondensator C in nF,?";C
120 IF C<0 THEN 110
130 INPUT "Eckfrequenz fg in Hz,?";F
140 IF F<0 THEN 130
150 IF F>0 THEN 190
160 IF R*C=0 THEN 70
170 F=5E5/(PI*R*C)
180 GOTO 250
190 IF R>0 THEN 230
200 IF C=0 THEN 70
210 X=0:R=5E5/(PI*F*C)
220 GOTO 250
230 IF C>0 THEN 70
240 C=5E5/(PI*R*F)
250 CLS:PRINT "R =";R;" k":X=X+42+(6*(LEN STR$(R))):
    GCURSOR(X,7):GPRINT OOS
260 PRINT "C =";C;" nF":WAIT:PRINT "fg =";F;" Hz->ENT."
280 GOTO 30
290 N=0
300 CLS:WAIT 0:PRINT "Filter 2. Ordnung":PRINT "1:
    Butterworth;2:Bessel 3:krit.Daempfung;4:sonst."
310 INPUT "Waehle Filtertyp<1-4>?";N
360 IF N<>1 THEN 410
370 CLS:PRINT "Butterworth-Filter"
380 A=1.4142:B=1:GOTO 590
410 IF N<>2 THEN 460

```

```

420 CLS:PRINT "Bessel-Filter"
430 A=1.3617:B=.618:GOTO 590
460 IF N<>3 THEN 510
470 CLS:PRINT "Filter mit krit.Daempf."
480 A=1.2872:B=.4142:GOTO 590
510 IF N<>4 THEN 310
520 CLS:PRINT "Filter mit folgenden Koeffizienten":
PRINT ""
530 INPUT "a > 0 ?";A
540 IF A<=0 THEN 530
560 INPUT "b > 0 ?";B
570 IF B<=0 THEN 560
590 PRINT "":INPUT "Eckfrequenz fg>0 in Hz?";F
600 IF F<=0 THEN 590
620 N=0:CLS:PRINT "":INPUT "1:Tiefpass, 2:Hochpass,?";N
640 IF N<>1 THEN 740
650 CLS:PRINT "":PRINT "Tiefpass":PRINT ""
660 INPUT "R = R1 = R2 in kOhm, ?";R
670 IF R<=0 THEN 660
671 CLS:GOSUB 20000
675 CLS:WAIT 0:PRINT "Tiefpass"
680 PRINT "R =";R;" k":X=41+(6*(LEN STR$(R))):
GCURSOR(X,15):GPRINT OO$
690 C=2.5E5*A/(PI*R*F)
700 PRINT "C1="";C;" nF"
710 C=1E6*B/(A*PI*F*R)
720 PRINT "C2="";C;" nF":X=41+(6*(LEN STR$(C))):
GCURSOR(X,31):GPRINT &08;&08;&2A;&1C;&08;&00;&3E;
725 WAIT:GPRINT &2A;&22;&00;&3E;&04;&08;&3E;&00;&20
730 GOTO 30
740 IF N<>2 THEN 620
750 CLS:PRINT "Hochpass":PRINT "":
INPUT "C = C1 = C2 in nF ?";C
760 IF C<=0 THEN 750
770 GOSUB 20100
775 CLS:WAIT 0:PRINT "Hochpass"
780 PRINT "C =";C;" nF"
790 R=1E6/(A*PI*F*C)
800 PRINT "R1="";R;" k":X=35+(6*(LEN STR$(R))):
GCURSOR(X,23):GPRINT OO$
810 R=2.5E5*A/(B*PI*F*C)
820 PRINT "R2="";R;" k":X=35+(6*(LEN STR$(R))):GCURSOR
(X,31):GPRINT OO$;&00;&08;&08;&2A;&1C;&08;&00;&3E;
825 WAIT:GPRINT &2A;&22;&00;&3E;&04;&08;&3E;&00;&20
830 REM **Ende der Ausgabe**
835 CLS:INPUT "Neue Filterberechn. ?=>F,
Inhaltsverzeichnis ?=>I WAEHLE !";Z$
840 IF Z$<>"F" AND Z$<>"I" THEN 835
845 IF Z$="F" THEN 30
850 IF Z$="I" THEN 10
855 REM **ENDE FILTERPROGRAMM**
1000 "F":CLEAR:PRINT=LPRINT:DIM AG(149)
1010 CLS:WAIT 0:PRINT "Wirkung aktiver Filter 1. bis
4. Ordnung"
1020 M=0
1030 INPUT "1>Tiefpass,2>Hochpass?";M
1040 IF M<>1 THEN 1070
1050 CLS:PRINT "Tiefpass"
1060 GOTO 1090

```

```

1070 IF M<>2 THEN 1020
1080 CLS:PRINT "Hochpass"
1090 N=0
1100 INPUT "Filterordnung?";N
1110 IF(N-1)*(N-3)<>0 THEN 1240
1120 CLS:PRINT "Filter 1.Ordnung.           Bauteile:"
1130 INPUT "R in Kiloohm?";R
1140 IF R<=0 THEN 1130
1150 INPUT "C in Nanofarad?";C
1160 IF C<=0 THEN 1150
1170 CLS:PRINT "R =";R;" kOhm":CURSOR(24):
      PRINT "C =";C;" nF"
1180 I=PI*R*C/500000
1190 IF M=1 THEN 1210
1200 I=-1/I
1210 IF N=1 THEN 1490
1220 GOSUB 1310
1230 GOTO 1490
1240 IF(N-2)*(N-4)<>0 THEN 1100
1250 GOSUB 1310
1260 H1=H2
1270 I1=I2
1280 IF N=2 THEN 1490
1290 GOSUB 1310
1300 GOTO 1490
1310 CLS:PRINT "Filter 2. Ordnung           Bauteile:"
1320 INPUT "R1 in kOhm?";R
1330 IF R<=0 THEN 1320
1340 INPUT "R2 in kOhm?";S
1350 IF S<=0 THEN 1340
1360 INPUT "C1 in nF?";C
1370 IF C<=0 THEN 1360
1380 INPUT "C2 in nF?";D
1390 IF D<=0 THEN 1380
1400 CLS:PRINT "R1=";R;" kOhm;R2=";S;" kOhm"
1410 CURSOR(48):PRINT "C1=";C;" nF ;C2=";D;" nF"
1420 H2=-PI*PI*C*D*R*S/2.5E11
1430 IF M=2 THEN 1460
1440 I2=PI*(R+S)*C/5E5
1450 RETURN
1460 H2=1/H2
1470 I2=-5E5*(C+D)/(PI*R*C*D)
1480 RETURN
1490 INPUT "Anfangsfrequenz in Hz?";F
1500 IF F<=0 THEN 1490
1510 INPUT "Endfrequenz in Hz?";E
1520 IF E<F THEN 1510
1530 INPUT "Schrittweite in Hz?";B
1540 IF B<=0 THEN 1530
1550 CLS:PRINT "Frequ. Amplitude  Wink."
1560 CURSOR(24):PRINT "in Hz  absolut  dB Grad"
1565 ASS=" " :GS=" " :H=5:F1=F
1570 C=(2-M)*F+(M-1)/F
1580 D=(2-M)*F*F+(M-1)/(F*F)
1590 J=C*I
1600 J1=C*I1
1610 J2=C*I2
1620 K1=1+D*H1
1630 K2=1+D*H2

```

```
1640 IF(N-1)*(N-3)<>0 THEN 1730
1650 X=1
1660 Y=J
1670 GOSUB 1890
1680 W=U
1690 Z=V
1700 IF N=1 THEN 1930
1710 GOSUB 1810
1720 GOTO 1930
1730 X=K1
1740 Y=J1
1750 GOSUB 1890
1760 W=U
1770 Z=V
1780 IF N=2 THEN 1930
1790 GOSUB 1810
1800 GOTO 1930
1810 P=U
1820 Q=V
1830 X=K2
1840 Y=J2
1850 GOSUB 1890
1860 W=U*P-V*Q
1870 Z=U*Q+P*V
1880 RETURN
1890 A=X*X+Y*Y
1900 U=X/A
1910 V=-Y/A
1920 RETURN
1930 A=SQR(W*W+Z*Z)
1940 P=20*LOG(A)/LOG(10):IF G$="J" LET H=H+1:AG(H)=P
1950 DEGREE
1960 Q=ATN(Z/W)
1970 IF W>=0 THEN 2020
1980 IF Z<0 THEN 2010
1990 Q=Q+180
2000 GOTO 2020
2010 Q=Q-180
2020 J=PI*Q/180:IF G$<>"J" GOTO 2030
2022 F=F+B:IF F>E THEN RETURN
2024 GOTO 1570
2030 CURSOR(48):WAIT 59:PRINT USING "*****&";F;ASS;USING
    "##.##&";A;ASS;USING "####.&";P;ASS;USING "####";Q
2050 WAIT:USING:F=F+B
2060 IF F>E GOSUB 2100
2070 GOTO 1570
2100 INPUT "Grafische Darst. J/N ?";G$
2110 IF G$="N" THEN 1490
2120 B=(E-F1)/143:F=F1
2130 GOSUB 1570
2140 GOSUB 20500
2150 GOTO 1490
9000 "A":CLEAR:CLS:PRINT=PRINT
9003 DIM AR(20),BI(20),C2(20),R1(20),R2(20),FIS(0)*40
9005 WAIT 0:PRINT "Dimensionierung aktiver Filter
    Einfachmitkopplung"
9006 WAIT 100:PRINT ""
9007 CLS:PRINT "Filterkarakteristik ?"
9008 CLS:WAIT 0:PRINT "Butterworth=1";CHR$ &7C;"Bessel=2
    Tschebyscheff=3";CHR$ &7C;"Kritisch"
```



```

9009 CURSOR(48):PRINT "Daempfung=4";CHR$ &7C;"Sonstige=5"
9010 INPUT "Waehle Filterart>1 - 5<";M
9012 IF M>5 OR M<1 GOTO 9008
9013 OG=10:IF M=1 LET OG=40
9015 CLS:PRINT "Ordnung ODG(1.bis ";OG;")":INPUT "?";ODG
9020 ODG=INT(ODG+.5):IF ODG<1 OR ODG>OG THEN 9015
9025 REM O$->Gerade=G,Ungerade=U
9030 O$="G":IF ODG/2-INT(ODG/2)>.1 LET O$="U"
9035 V=1
9040 IF O$="G" THEN 9055
9045 CLS:INPUT "Verst.des Filt.(X-fach)?";V
9050 V=ABS(V):IF V<1 THEN 9045
9055 REM **Ber.der OP-Anzahl**
9060 N=INT(ODG/2):IF O$="U" LET N=N+1
9065 INPUT "Hochpass=H;Tiefpass=T?";A$
9070 IF A$<>"H" AND A$<>"T" THEN 9065
9075 CLS:INPUT "Grenzfrequenz FG in Hz?";FG:
    IF FG<=0 THEN 9075
9080 ON M GOSUB 9800,9840,9865,9920,9945
9085 CLS:INPUT "Kondensator C1 in nF?";C1
9090 C1=C1/1E9
9095 IF C1<=0 THEN 9080
9175 IF A$="H" THEN 9260
9180 REM **Ber.&Eing.C2 beim TP**
9182 IF ODG=1 THEN 9250
9185 XB=1:IF O$="U" LET XB=2
9195 FOR I=XB TO N
9200 CX=4*BI(I)*C1/(AR(I)*AR(I))
9201 CX=INT(CX*1E12+1)/1E3
9205 CLS:PRINT "Beim ";I;".OP C2(in nF)so"
9210 CURSOR(24):WAIT 240:PRINT "eingeben,dass C2 nahe
    CXliegt und C2>=CX ist !"
9225 CLS:WAIT 0:PRINT "Richtwert CX= ";CX;"nF"
9230 INPUT "C2      =?";C2
9235 IF C2<CX THEN PAUSE "C2 zu klein!Neueingabe!":
    GOTO 9205
9240 C2(I)=C2/1E9
9245 NEXT I
9250 REM Wenn ODG=1,dann entf.C2
9255 IF A$="T" THEN 9330
9260 REM **C2 beim Hochpass**
9265 FOR I=1 TO N
9270 C2(I)=C1
9275 NEXT I
9280 REM **R1&R2 beim 1.HP/Ungr.**
9285 R1(1)=1/(2*PI*FG*AR(1)*C1):X=R1(1):GOSUB 10000
9287 R1(1)=X
9290 R2(1)=R1(1)*V
9292 IF ODG=1 THEN 9325
9295 REM **R1&R2 beim HP**
9300 XB=1:IF O$="U" LET XB=2
9305 FOR I=XB TO N
9310 R1(I)=1/(PI*FG*C1*AR(I)):X=R1(I):GOSUB 10000
9312 R1(I)=X
9315 R2(I)=AR(I)/(4*PI*FG*C2(I)*BI(I)):X=R2(I):
    GOSUB 10000
9317 R2(I)=X
9320 NEXT I
9325 GOTO 9385

```

```

9330 REM **R1&R2 beim 1.TP/Ungr.**
9335 R2(1)=AR(1)/(2*PI*FG*C1):X=R2(1):GOSUB 10000
9337 R2(1)=X
9340 R1(1)=R2(1)/V
9342 IF ODG=1 THEN 9385
9345 REM **R1&R2 beim TP**
9350 XB=1:IF OS="U" LET XB=2
9355 FOR I=XB TO N
9360 D=AR(I)*AR(I)*C2(I)*C2(I)-4*BI(I)*C1*C2(I)
9365 NE=4*PI*FG*C1*C2(I)
9370 R1(I)=(AR(I)*C2(I)-SQR(D))/NE:X=R1(I):GOSUB 10000
9372 R1(I)=X
9375 R2(I)=(AR(I)*C2(I)+SQR(D))/NE:X=R2(I):GOSUB 10000
9377 R2(I)=X
9380 NEXT I
9385 REM **AUSGABE**
9386 C1=C1*1E9
9387 IF OS="U" LET RF=R1(1)
9388 IF OS="G" AND AS="T" LET RF=R1(1)+R2(1)
9389 IF OS="G" AND AS="H" LET RF=R1(1)
9390 PRINT=LPRINT:REM **GESAMTDATEN**
9397 CLS:WAIT 0:PRINT "GESAMTFILTER:":CURSOR(24):
PRINT "Schaltung =":IF AS="H"
THEN PRINT " Hochpass"
9410 IF AS="T" THEN PRINT " Tiefpass"
9412 CURSOR(72):WAIT 100:PRINT FIS(0)
9415 CLS:WAIT 0:PRINT "Odg=";ODG;":Gfr=";FG;"Hz"
9425 CURSOR(24):PRINT "OP-Anz.=";N;":Verst.=";V
9430 CURSOR(48):PRINT "Rin>";RF;"Ohm"
9435 CURSOR(72):WAIT:PRINT "Daempfung=";ODG*6;"dB/O:":
ODG*20;"dB/D"
9455 IF OS="G" THEN 9505
9470 IF AS="H" GOSUB 20300
9475 IF AS="T" GOSUB 20200
9477 CLS:WAIT 0:PRINT "OP-Nr.= 1"
9480 CURSOR(24):PRINT "R1 =";R1(1);"Ohm"
9485 CURSOR(48):PRINT "R2 =";R2(1);"Ohm"
9490 CURSOR(72):WAIT:USING:PRINT "C1 =";C1;"nF"
9500 IF N=1 THEN 9570
9505 REM **Ausg.Filter 2.Ordn.**
9510 XB=1:IF OS="U" LET XB=2
9512 FOR I=XB TO N
9514 C2(I)=C2(I)*1E9
9525 IF AS="H" GOSUB 20100
9530 IF AS="T" GOSUB 20000
9535 CLS:WAIT 0:PRINT "OP-Nr.=";I
9540 CURSOR(24):PRINT "R1 =";R1(I);"Ohm"
9545 CURSOR(48):PRINT "R2 =";R2(I);"Ohm"
9550 CURSOR(72):WAIT:PRINT "C1/C2 =";C1;"nF/";C2(I);"nF"
9565 NEXT I
9570 REM **Ende der Ausgabe**
9580 CLS:INPUT "Neue Filterberechn.?=>F,
Inhaltsverzeichnis ?=>I WAEHLE !";Z$
9600 IF Z$<>"F" AND Z$<>"I" THEN 9580
9605 IF Z$="F" THEN 9000
9610 IF Z$="I" THEN "I"
9620 REM **ENDE FILTERPROGRAMM**
9701 DATA 1,0

```

9702 DATA 1.2872,.4142
9703 DATA.5098,0,1.0197,.2599
9704 DATA.87,.1892,.87,.1892
9705 DATA.3856,0,.7712,.1487,.7712,.1487
9706 DATA.6999,.1225,.6999,.1225,.6999,.1225
9707 DATA.3226,0,.6453,.1041,.6453,.1041,.6453,.1041
9708 DATA.6017,.0905,.6017,.0905,.6017,.0905,.6017,
.0905
9709 DATA.2829,0,.5659,.0801,.5659,.0801,.5659,.0801,
.5659,.0801
9710 DATA.5358,.0718,.5358,.0718,.5358,.0718,.5358,
.0718,.5358,.0718
9711 DATA 1,0
9712 DATA 1.3617,.618
9713 DATA.756,0,.9996,.4772
9714 DATA 1.3397,.4889,.7743,.389
9715 DATA.6656,0,1.1402,.4128,.6216,.3245
9716 DATA 1.2217,.3887,.9686,.3505,.5131,.2756
9717 DATA.5937,0,1.0944,.3395,.8304,.3011,.4332,.2381
9718 DATA 1.1112,.3162,.9754,.2979,.7202,.2621,.3728,
.2087
9719 DATA.5386,0,1.0244,.2834,.871,.2636,.632,.2311,
.3257,.1854
9720 DATA 1.0215,.265,.9393,.2549,.7815,.2351,.5604,
.2059,.2883,.1665
9721 DATA 1,0
9722 DATA 1.3614,1.3827
9723 DATA 1.8636,0,.6402,1.1931
9724 DATA 2.6282,3.4341,.3648,1.1509
9725 DATA 2.9235,0,1.3025,2.3534,.229,1.0833
9726 DATA 3.8645,6.9797,.7528,1.8573,.1589,1.0711
9727 DATA 4.0211,0,1.8729,4.1795,.4861,1.5676,.1156,
1.0443
9728 DATA 5.1117,11.9607,1.0639,2.9365,.3439,1.4206,
.0885,1.0407
9729 DATA 5.1318,0,2.4283,6.6307,.6839,2.2908,.2559,
1.3133,.0695,1.0272
9730 DATA 6.3648,18.3695,1.3582,4.3453,.4822,1.944,
.1994,1.252,.0563,1.0263
9731 DATA 1,0
9732 DATA 1.3022,1.5515
9733 DATA 2.2156,0,.5442,1.2057
9734 DATA 2.5904,4.1301,.3039,1.1697,
9735 DATA 3.5711,0,1.128,2.4896,.1872,1.0814
9736 DATA 3.8437,8.5529,.6292,1.9124,.1296,1.0766
9737 DATA 4.952,0,1.6338,4.4899,.3987,1.5834,.0937,
1.0423
9738 DATA 5.1019,14.7608,.8916,3.0426,.2806,1.4334,
.0717,1.0432
9739 DATA 6.3415,0,2.1252,7.1711,.5624,2.3278,.2076,
1.3166,.0562,1.0258
9740 DATA 6.3634,22.7468,1.1399,4.5167,.3939,1.9665,
.1616,1.2569,.0455,1.0277
9741 DATA 1,0
9742 DATA 1.1813,1.7775
9743 DATA 2.7994,0,.43,1.2036
9744 DATA 2.4025,4.9862,.2374,1.1896
9745 DATA 4.6345,0,.909,2.6036,.1434,1.075
9746 DATA 3.588,10.4648,.4925,1.9622,.0995,1.0826

```

9747 DATA 6.476,0,1.3258,4.7649,.3067,1.5927,.0714,
1.0384
9748 DATA 4.7743,18.151,.6991,3.1353,.2153,1.4449,
.0547,1.0461
9749 DATA 8.3198,0,1.7299,7.658,.4337,2.3549,.1583,
1.3174,.0427,1.0232
9750 DATA 5.9618,28.0376,.8947,4.6644,.3023,1.9858,
.1233,1.2614,.0347,1.0294
9751 DATA 1,0
9752 DATA 1.065,1.9305
9753 DATA 3.3496,0,.3559,1.1923
9754 DATA 2.1853,5.5339,.1964,1.2009
9755 DATA 5.6334,0,.762,2.653,.1172,1.0686
9756 DATA 3.2721,11.6773,.4077,1.9873,.0815,1.0861
9757 DATA 7.9064,0,1.1159,4.8963,.2515,1.5944,.0582,
1.0348
9758 DATA 4.3583,20.2948,.5791,3.1808,.1765,1.4507,
.0448,1.0478
9759 DATA 10.1759,0,1.4585,7.8971,.3561,2.3651,.1294,
1.3165,.0348,1.021
9760 DATA 5.4449,31.3788,.7414,4.7363,.2479,1.9952,
.1008,1.2638,.0283,1.0304
9800 FIS(0)="Butterworth-Filter":RADIAN:REM **Ber.der
Filterkonst.**
9805 FOR I=1 TO N
9810 BI(I)=1
9815 IF OS="G" LET AR(I)=2*COS((2*I-1)*PI/(2*ODG))
9820 IF OS="U" LET AR(I)=2*COS((I-1)*PI/ODG)
9825 NEXT I
9830 IF OS="U" LET AR(1)=1:BI(1)=0
9835 RETURN
9840 FIS(0)="Bessel-Filter":REM **Filterkonst.-BESSEL-**
9845 RESTORE(9710+ODG):FOR I=1 TO N
9850 READ AR(I),BI(I)
9855 NEXT I
9860 RETURN
9865 FIS(0)="Tschebyscheff-Filter ":REM **Filterkonst.
-TSCHEBYSCHJEFF-**
9870 CLS:INPUT "Tschebyscheff - Filter Welligkeit in dB
. 0.5 ; 1 ; 2 ; 3 ?";W
9880 IF W=.5 LET W=0:TSS="0.5dB Welligkeit":GOTO 9890
9882 IF W=1 LET W=10:TSS="1 dB Welligkeit":GOTO 9890
9884 IF W=2 LET W=20:TSS="2 dB Welligkeit":GOTO 9890
9886 IF W<>3 GOTO 9870
9888 W=30:TSS="3 dB Welligkeit"
9890 FIS(0)=FIS(0)+TSS:RESTORE(9720+W+ODG):FOR I=1 TO N
9895 READ AR(I),BI(I)
9900 NEXT I
9910 RETURN
9920 FIS(0)="Filter mit kritischer Daempfung:
REM**Filterkonst.-KRITISCHE DAEMP
9925 RESTORE(9700+ODG):FOR I=1 TO N
9930 READ AR(I),BI(I)
9935 NEXT I
9940 RETURN
9945 FIS(0)="Filter mit frei gewaehl-ten Konstanten":
REM **Filterkonst.-SONSTIGE-**
9950 CLS:PRINT "Filter mit folgenden Koeffizienten:"

```

```
9955 FOR I=1 TO N:PRINT "a(";I;" ) > 0 ! b(";I;" ) > 0 !":
      INPUT "a=?";AR(I),"b=?";BI(I)
9960 NEXT I
9965 RETURN
10000 Y=LEN STR$(INT X):IF Y>3 LET X=INT X
10010 RETURN
20000 CLS:PRINT=PRINT:WAIT 0:CURSOR(9):
      PRINT "Tiefpass 2.Ordn":CURSOR(33):
      PRINT "12 dB/Oktave"
20002 CURSOR(57):PRINT "Weiter mit":CURSOR(81):
      PRINT "ENTER !"
20003 LINE(11,1)-(46,1):LINE-(46,25):LINE-(28,25):
      LINE-(28,16):LINE-(31,16)
20005 LINE(50,13)-(41,13):LINE-(32,22):LINE-(32,4):
      LINE-(40,12):LINE(35,9)-(35,11)
20010 LINE(34,10)-(36,10):LINE(34,16)-(36,16):
      LINE(18,10)-(31,10):LINE(22,11)-(22,18)
20015 LINE(20,19)-(24,19):LINE(20,21)-(24,21):
      LINE(22,22)-(22,29):LINE(3,30)-(50,30)
20020 LINE(14,9)-(18,9):LINE(14,11)-(18,11):
      LINE(11,2)-(11,3):LINE(9,4)-(13,4):
      LINE(9,6)-(13,6)
20025 LINE(11,7)-(11,8):LINE(10,9)-(12,9):
      LINE(8,10)-(14,10):LINE(10,11)-(12,11)
20030 LINE(21,9)-(23,9):LINE(21,31)-(23,31):
      LINE(4,9)-(8,9):LINE(4,11)-(8,11):
      LINE(3,10)-(4,10)
20035 PSET(49,12):PSET(49,14):PSET(49,29):PSET(49,31):
      PSET(45,12):PSET(45,14):PSET(47,12):PSET(47,14)
20040 PSET(21,11):PSET(23,11):PSET(21,29):PSET(23,29):
      PSET(1,30):PSET(2,29):PSET(2,31):PSET(1,10):
      PSET(2,9)
20045 PSET(2,11):GCURSOR(3,7):GPRINT &F8;&88;&00;&E8;&98
      :GCURSOR(4,20):GPRINT &1F;&0D;&12;&00;&02;&1F
20050 GCURSOR(14,20):GPRINT &DF;&4D;&12;&80;&DD;&13:
      GCURSOR(14,28):WAIT:GPRINT &07;&04;&00;&00;&07
20060 PRINT=LPRINT:RETURN
20100 CLS:PRINT=PRINT:WAIT 0:CURSOR(9):
      PRINT "Hochpass 2.Ordn":CURSOR(33):
      PRINT "12dB/Oktave":CURSOR(57):PRINT "Weiter mit"
20102 CURSOR(81):PRINT "ENTER !"
20105 LINE(11,3)-(11,1):LINE-(46,1):LINE-(46,25):
      LINE-(28,25):LINE-(28,16):LINE-(31,16)
20110 LINE(50,13)-(41,13):LINE-(32,22):LINE-(32,4):
      LINE-(40,12):LINE(35,9)-(35,11)
20115 LINE(34,10)-(36,10):LINE(34,16)-(36,16):
      LINE(18,10)-(31,10):LINE(22,11)-(22,18)
20120 LINE(23,18)-(23,22):LINE(21,18)-(21,22):
      LINE(22,22)-(22,29):LINE(3,30)-(50,30)
20125 LINE(15,8)-(15,12):LINE(17,8)-(17,12):
      LINE(10,3)-(10,7):LINE(12,3)-(12,7)
20130 LINE(11,7)-(11,8):LINE(10,9)-(12,9):
      LINE(8,10)-(14,10):LINE(10,11)-(12,11)
20135 LINE(21,9)-(23,9):LINE(21,31)-(23,31):
      LINE(5,8)-(5,12):LINE(7,8)-(7,12):
      LINE(3,10)-(4,10)
20140 PSET(49,12):PSET(49,14):PSET(49,29):PSET(49,31):
      PSET(45,12):PSET(45,14):PSET(47,12):PSET(47,14)
```

```
20145 PSET(21,11):PSET(23,11):PSET(21,29):PSET(23,29):
      PSET(1,30):PSET(2,29):PSET(2,31):PSET(1,10):
      PSET(2,9)
20150 PSET(2,11):GCURSOR(3,6):GPRINT &F8;&68;&90;&00;&E8
      ;&98:GCURSOR(4,21):GPRINT &1F;&11;&00;&02;&1F
20152 GCURSOR(14,21)
20155 GPRINT &DF;&51;&80;&1D;&93;&C0:GCURSOR(14,29):
      WAIT:GPRINT &07;&03;&04;&00;&00;&07
20160 PRINT=LPRINT:RETURN
20200 CLS:PRINT=PRINT:WAIT 0:CURSOR(9):
      PRINT "Filter 1.Ordn.":CURSOR(33):
      PRINT "Tiefpass6dB/Okt"
20202 CURSOR(57):PRINT "Weiter mit":CURSOR(81):
      PRINT "ENTER !"
20205 LINE(11,3)-(11,1):LINE-(46,1):LINE-(46,12):
      LINE(50,13)-(41,13):LINE-(32,22):LINE-(32,4)
20210 LINE-(40,12):LINE(35,15)-(35,17):
      LINE(34,10)-(36,10):LINE(34,16)-(36,16):
      LINE(4,9)-(8,9)
20215 LINE(31,16)-(28,16):LINE-(28,29):
      LINE(31,10)-(8,10):LINE(4,11)-(8,11):
      LINE(10,9)-(12,9)
20220 LINE(10,11)-(12,11):LINE(9,4)-(13,4):
      LINE(9,6)-(13,6):LINE(21,11)-(23,11)
20225 LINE(21,9)-(23,9):LINE(22,8)-(22,7):
      LINE(22,3)-(22,2):LINE(21,0)-(23,0):
      LINE(21,3)-(21,7)
20230 LINE(23,3)-(23,7):LINE(3,10)-(4,10):
      LINE(3,30)-(50,30):LINE(27,31)-(29,31)
20235 LINE(45,14)-(47,14):PSET(49,12):PSET(49,14):
      PSET(49,29):PSET(49,31):PSET(45,12):PSET(47,12)
20240 PSET(27,29):PSET(29,29):PSET(1,30):PSET(2,29):
      PSET(2,31):PSET(1,10):PSET(2,9):PSET(2,11)
20242 LINE(11,7)-(11,8):GCURSOR(3,6)
20245 GPRINT &F8;&88;&00;&10;&F8:GCURSOR(4,21):
      GPRINT &1F;&0D;&12;&00;&02;&1F:GCURSOR(25,8)
20250 WAIT:GPRINT &F9;&69;&91;&01;&E9;&99
20260 PRINT=LPRINT:RETURN
20300 CLS:PRINT=PRINT:WAIT 0:CURSOR(9):
      PRINT "Filter 1.Ordn.":CURSOR(33):
      PRINT "Hochpass6dB/Okt"
20302 CURSOR(57):PRINT "Weiter mit":CURSOR(81):
      PRINT "ENTER !"
20305 LINE(22,1)-(46,1):LINE-(46,12):LINE(50,13)-(41,13)
      :LINE-(32,22):LINE-(32,4):LINE-(40,12)
20310 LINE(35,15)-(35,17):LINE(34,10)-(36,10):
      LINE(34,16)-(36,16):LINE(4,9)-(8,9)
20315 LINE(4,11)-(8,11):LINE(31,16)-(28,16):
      LINE-(28,29):LINE(31,10)-(18,10):
      LINE(8,10)-(14,10)
20320 LINE(17,8)-(17,12):LINE(15,8)-(15,12):
      LINE(21,11)-(23,11):LINE(21,9)-(23,9)
20325 LINE(22,8)-(22,7):LINE(22,3)-(22,2):
      LINE(21,3)-(21,7):LINE(23,3)-(23,7)
20330 LINE(3,10)-(4,10):LINE(3,30)-(50,30):
      LINE(27,31)-(29,31):LINE(45,14)-(47,14):
      PSET(49,12)
20335 PSET(49,14):PSET(49,29):PSET(49,31):PSET(45,12):
      PSET(47,12):PSET(27,29):PSET(29,29):PSET(1,30)
```

```
20340 PSET(2,29):PSET(2,31):PSET(1,10):PSET(2,9):
      PSET(2,11):GCURSOR(4,21)
20345 GPRINT &1F;&0D;&12;&00;&02;&1F:GCURSOR(25,8):
      GPRINT &F9;&69;&91;&01;&E9;&99:GCURSOR(14,21)
20350 WAIT:GPRINT &1F;&11;&00;&02;&1F
20360 PRINT=LPRINT:RETURN
20500 CLS:WAIT 0:PSET(4,1):PSET(6,1):LINE(5,0)-(5,31):
      LINE(4,30)-(149,30):PSET(148,29):PSET(148,31)
20505 PSET(4,6):PSET(4,12):PSET(4,18):PSET(4,24):
      LINE(6,6)-(149,6),&2222:FOR I=1 TO 10
20506 GCURSOR(0,7):GPRINT &EO;&A0;&EO:
      LINE(I*14+6,31)-(I*14+6,0),&0101:
      PSET(I*14+6,31):NEXT I
20510 FOR I=6 TO 149
20515 AG(I)=INT(AG(I)+.5)
20520 A=6+(-2*AG(I))
20530 PSET(I,A),X
20540 NEXT I
20542 DIM Z$(0)*5:IF M=1 THEN GCURSOR(6,29):GOTO 20544
20543 GCURSOR(6,7)
20544 Z$(0)=STR$ F1:GOSUB 31000
20545 IF M=1 THEN GCURSOR(128,7):GOTO 20547
20546 GCURSOR(128,29)
20547 Z$(0)=STR$ E:GOSUB 31000
20550 WAIT:GPRINT
20560 RETURN
31000 DATA 0,0,0,0,111,0,7,0,7,20,127,20,44,127,26
31001 DATA 50,8,38,118,73,126,4,2,1,28,34,65,65,34,28
31002 DATA 42,28,42,8,62,8,0,128,96,8,8,8,0,96,96
31003 DATA 96,28,3,62,65,62,4,6,127,114,73,70,34,73,54
31004 DATA 15,8,126,47,73,49,60,74,49,1,121,7,54,73,54
31005 DATA 38,73,62,0,36,36,0,128,100,24,36,66,20,20,20
31006 DATA 66,36,24,2,89,6,62,77,46,124,11,124,127,73,54
31007 DATA 62,65,34,127,65,62,127,73,65,127,9,1,62,73,58
31008 DATA 127,8,127,65,127,65,32,64,63,127,28,99,127,64
      ,64
31009 DATA 127,14,127,127,28,127,127,65,127,127,9,6,62,97,126
31010 DATA 127,9,118,70,73,49,1,127,1,127,64,127,63,96
      ,63
31011 DATA 127,48,127,119,8,119,15,120,15,113,73,71,127
      ,65,65
31012 DATA 126,73,98,65,65,127,2,127,2,128,128,128,4,2,1
31013 DATA 32,84,120,126,72,48,56,68,40,48,72,126,56,84
      ,88
31014 DATA 8,126,10,72,84,60,127,4,120,0,122,64,64,128
      ,122
31015 DATA 127,16,108,126,64,0,124,24,124,124,4,120,56
      ,68,56
31016 DATA 252,36,24,24,36,252,120,4,4,72,84,36,4,127,68
31017 DATA 60,64,124,60,96,60,124,48,124,108,16,108,156
      ,96,60
31018 DATA 100,84,76,16,16,16,0,0,255,16,16,255,8,28,8
31019 WAIT 0
31020 FOR TT=1 TO LEN Z$(0)
31030 ZZ=ASC(MID$(Z$(0),TT,1))
31040 ZZ=ZZ-32
31050 RESTORE 31000+INT(ZZ/5)
31055 IF INT(ZZ/5)=ZZ/5 THEN 31090
31060 FOR DD=1 TO((ZZ/5-INT(ZZ/5))*5)*3
```

```

31070 READ AA
31080 NEXT DD
31090 FOR DD=0 TO 2
31100 READ AA
31110 GPRINT AA;
31120 NEXT DD
31130 GPRINT 0;
31135 IF TT=37 THEN GPRINT
31140 NEXT TT
31150 RETURN

```

Reisekonto

Beschreibung:

Leistung:

Das Programm führt eine "Reisekasse" für mehrere Personen in mehreren Währungen. Allgemeiner kann jede Buchhaltung mit Haben/Soll-Einträgen für verschiedene Einheiten (Währungen) und verschiedene Posten (Personen) realisiert werden. Die Umrechnungsfaktoren zwischen den Währungen können jederzeit rückwirkend geändert werden. Einträge und Abrechnungen können elegant für beliebige Untergruppen der Personen getätigt werden.

Implementierung (Initialisierung):

Niemals darf RUN eingegeben werden. RUN zerstört immer den Speicherinhalt und kann leider nicht abgefangen werden. RUN ist demnach die größte Gefahr für die Daten und wird deshalb nicht zum initialisieren genutzt.

Initialisierung	Prompt	Bedeutung
DEF ,(Komma)	*Pers ->	max.Anzahl der Personen (>1)
	*Curr ->	max.Anzahl der Währungen (>1)

Reicht der Speicher nicht, tritt ein Fehler auf. Es sollte vorsichtshalber ein etwas zu großer Wert gewählt werden, um flexibel für spätere Änderungen zu sein.

Bedienung:

Def-Taste	Prompt	Eingabe	Bedeutung
DEF ,(Komma)		siehe Initialisierung	
DEF M	keiner	keine	Save Datenbestand auf Kassette
DEF SPC(Leerzeichen)	"	keine	Lade Datenbestand von Kassette
DEF J	Define ->	PASS=abcd	Setze Passwort zur Zeichenkette
		'abcd'. Eine Eingabe von 4 Leer-	
		zeichen ' ' schaltet den	
		Passwortschutz wieder aus.	
		Die vier Passwortzeichen sind	
		bei der Passwortabfrage bei	
		späterer Nutzung langsam in	
		Abständen von etwa 1/2 Sekunde	
		einzugeben.	
		BEISPIEL: DEFINE -> PASS=CODE	
	a..z		Eine Eingabe eines oder mehrerer
		Buchstaben definiert JEDEN	
		Buchstaben als Kürzel für genau	
		eine Person. Bereits definierte	
		Kürzel bleiben erhalten!	
		BEISPIEL: DEFINE -> NKWD	


```

| für Natascha Kurt Willi Domino
| *=a..z | Definiert das Zeichen '*' als
|         | Abkürzung für die Zeichenfolge
|         | 'a..z'. Genauer steht * für eine
|         | Teilmenge an Personen.
|         | BEISPIEL: DEFINE -> *=ND
DEF V |Currency -> |ab=<zahl> | Führt die beiden Zeichen als
|         | Währungskürzel ein und gibt ihr
|         | den Wert <zahl>. Dies kann auch
|         | benutzt werden, um den Wert neu
|         | festzulegen.
|         | BEISPIEL: CURRENCY -> DM=1
|         |           CURRENCY -> U$=1.695
|         | Ein Wert von Null zeigt eine
|         | nicht konvertierbare Einheit an,
|         | z.B. LT für Liter Super Bleifrei.
DEF H |?Currency? ->| ab
|         | Umrechnung.
|         | Der Wert der Anzeige, oder falls
|         | diese leer ist, die letzte Eingabe,
|         | wird als Betrag in der Arbeits-
|         | währung interpretiert und in die
|         | eingegebene Währung umgerechnet.
|         | Bei einem 2. ENTER wird diese zur
|         | neuen Arbeitswährung und durch
|         | einen Pieps wird dies angemerkt.
|         | Bei wiederholter Umrechnung kann
|         | die Währungseingabe unterbleiben.
|         | BEISPIEL (Arbeitswährung sei DM):
|         | Anzeige zeige      350.00
|         | DEFH ?CURRENCY? -> U$ <ENTER>
|         | Anzeige zeigt      206.49 U$
|         |
|         | Anzeige zeige      500.00
|         | DEF H ?CURRENCY? -> <ENTER>
|         | Anzeige zeigt      294.99 U$
|         |                   <ENTER>
|         | Anzeige zeige nun  500.00
|         | DEF H ?CURRENCY? -> <ENTER>
|         | Anzeige zeigt      500.00 U$
|         | denn U$ wurde zur Arbeitswährung.
DEF N |Account -> | a>"set"=(bc)<zahl> | Konto-Eintrag
|         | Die Person a
|         | zahlt für "set"
|         | in der Währung bc
|         | den Betrag <zahl>.
|         | (..) bedeutet, daß die Angabe
|         | von bc weggelassen werden darf,
|         | was einer Angabe der Arbeits-
|         | währung entspricht. Die Währung
|         | bc wird neue Arbeitswährung.
|         | "set" ist dabei von der Form:
|         | (*)d..g(-f..z)
|         | Beispiele für "set" sind:
|         | N, *W, *W-N mit der Bedeutung
|         | (vgl. DEF J) Natascha,
|         | (Natascha+Domino)+Willi,
|         | (Natascha+Domino)+Willi-Natascha
|         | = Domino+Willi.
|         | Eine falsche Eingabe wird gelöscht

```


Obwohl ich das Programm nur für den privaten Gebrauch schrieb, ist die Zuverlässigkeit dieser Reisekasse ansonsten sehr hoch, denn alle Beteiligten können leicht auf dem Laufenden gehalten werden, kein "Hammer" beim Abrechnen...

Werden ALLE Ausgaben eingegeben, ist die Geldmenge jeder Person bekannt ein Taschenrechner findet sich schneller wieder, als ein Zettel... Ein Batteriewechsel ist vermeidbar. (Falls nicht, muss der Gesamtkontostand gerettet werden. (Oder mit Kassette Save-Load)) keine Rechen-, höchstens Eingabefehler. Ein Speicherüberlauf bei langen Reisen kommt nicht vor. eventuelle Programmerrrors kann man getrost ignorieren, denn solange es nicht piepst, sind die internen Daten unverändert!

Speicherbedarf: Ein 1401 mit 3,5K reicht für etwa 6 Personen bei 4 Währungen. Mit einem 1403 und über 30K kann man schon eine Welt-Busreise abrechnen.

Listing:

```

10 REM R E I S E K O N T O
20 REM SHARP PC 140X
30 REM PUBLIC DOMAIN PROGRAM BY FALK LANGHAMMER
40 REM REFERENCE ADRESS : F.LANGHAMMER
50 REM . VENNSTR. 124
60 REM . D - 4000 DUESSELDORF 12 (W.GERMANY)
70 REM
80 REM BOOKKEEPING PROGRAM FOR N TRAVELLING PERSONS
  IN M COUNTRIES
90 REM =====
  =====
100 REM REMOVE THESE REM-LINES TO SAVE MEMORY-SPACE
110 PRINT "NEVER USE RUN !!"
120 BEEP 5:END
130 ", " GOSUB "CH"
140 IF ZZ=1 INPUT "DELETE OLD Y/N?";IN$:IF IN$<>"Y" END
150 CLEAR:INPUT "#PERS ->";MP:INPUT "#CURR ->";MW:
  MP=MP-1:MW=MW-1:GOSUB "IL"
160 NW=-1:NP=-1:PW$(0)=" "
170 BEEP 2:END
180 "N" GOSUB "CH"
190 "N1" INPUT "NEW ENTRY ->";IN$:GOSUB "IS"
200 PRINT USING "##";AN;USING "*****.##";
  LE;" ";W$(IW)
210 IF AN<=0 OR PJ=-1 END
220 HA(IW,PJ)=HA(IW,PJ)+LE:NE(PJ)=NE(PJ)+1:IF LE<0
  LET NE(PJ)=NE(PJ)-2
230 FOR I=0 TO NP:IF PO(I)=1 LET SO(IW,I)=SO(IW,I)+LE/AN
240 NEXT I:BEEP 1:PRINT "OK:"+IN$:GOTO "N1"
250 "V" GOSUB "CH":INPUT "CURRENCY ->";IN$:WSS(0)=IN$:
  GOSUB "WN"
260 IF WN=-1 LET NW=NW+1:W$(NW)=WSS(0):WN=NW
270 EX(WN)=VAL(MID$(IN$,4,11)):BEEP 1:END
280 "K" GOSUB "CH"
290 "K3" INPUT "ACCOUNT ->";IN$:Z=0:AD=1:GOSUB "NZ"
300 U=IW:V=IW:PSS(0)=MID$(IN$,Z,1):T2$(0)=" "
310 "K1"LE=0:FOR I=0 TO NP:IF PO(I)=0 THEN "K2"
320 FOR T=U TO V
330 F=1:IF EX(IW)<>0 LET F=EX(T)/EX(IW)
340 IF PSS(0)="*" AND WI=W1 LET LE=LE+NE(I):TES(0)="*"
350 IF PSS(0)="+" LET LE=LE+HA(T,I)*F:TES(0)="P"
360 IF PSS(0)="-" LET LE=LE+SO(T,I)*F:TES(0)="N"

```

```

370 IF PSS(0)="=" LET LE=LE+(HA(T,I)-SO(T,I))*F:
    TES(0)="S"
380 NEXT T
390 "K2" NEXT I
400 IF PSS(0)="#" PRINT TES(0)+T2$(0);USING "*****";
    LE;" ENTRIES":GOTO "K3"
410 PRINT TES(0)+T2$(0);USING "*****.##";
    LE+.005;" ";W$(IW)
420 IF T2$(0)="T" THEN "K3"
430 U=0:V=NW:T2$(0)="T":GOTO "K1"
440 "J" GOSUB "CH":INPUT "DEFINE ->";IN$
450 IF LEFT$(IN$,4)="PASS" LET PA=1:
    PWS(0)=MID$(IN$,6,4):GOTO "J1"
460 IF LEFT$(IN$,1)="*" LET Z=2:AD=1:GOSUB "NZ":
    FOR I=0 TO NP:SY(I)=PO(I):NEXT I:BEEP 1:END
470 Z=0
480 "J2"Z=Z+1:PSS(0)=MID$(IN$,Z,1):
    IF PSS(0)="" OR PSS(0)=":" BEEP 1:END
490 GOSUB "PN":IF PN=-1 LET NP=NP+1:P$(NP)=PSS(0)
500 GOTO "J2"
510 "J1" IF PWS(0)=" " LET PA=0
520 BEEP 1:END
530 "H" AREAD L2:IF L2=0 LET L2=LE
540 INPUT "?CURRENCY? ->";W$(0):GOSUB "WN":IF WN=-1 END
550 IF EX(WN)=0 PRINT "NO CONVERSION":L2=0:GOTO "H1"
560 L2=L2*EX(IW)/EX(WN)
570 "H1" PRINT USING "*****.##";L2+.005;" ";W$(WN):
    IW=WN:LE=L2:BEEP 1:END
580 "IS"PSS(0)=IN$:GOSUB "PN":PJ=PN:IF PJ=-1 THEN "ER"
590 Z=2:AD=1:GOSUB "NZ"
600 AN=0:FOR I=0 TO NP:AN=AN+PO(I):NEXT I
610 W$(0)=MID$(IN$,Z,2):GOSUB "WN"
620 IF WN<>-1 LET IW=WN:Z=Z+2
630 LE=VAL(MID$(IN$,Z,(17-Z))):RETURN
640 "NZ" FOR I=0 TO NP:PO(I)=0:NEXT I
650 "NW"Z=Z+1:PSS(0)=MID$(IN$,Z,1)
660 IF PSS(0)="*" FOR I=0 TO NP:PO(I)=SY(I):NEXT I:
    GOTO "NW"
670 IF PSS(0)="-" LET AD=0:GOTO "NW"
680 IF PSS(0)="=" LET Z=Z+1:RETURN
690 IF PSS(0)="" RETURN
700 GOSUB "PN":IF PN<>-1 LET PO(PN)=AD:GOTO "NW"
710 "ER" BEEP 3:PRINT "! ILLEGAL NAME !":END
720 "PN"PN=-1:FOR I=0 TO NP:IF PSS(0)=P$(I) LET PN=I
730 NEXT I:RETURN
740 "WN"WN=-1:FOR I=0 TO NW:IF W$(0)=W$(I) LET WN=I
750 NEXT I:RETURN
760 "CH" IF PA=0 RETURN
770 WAIT 64:PRINT "ENTER PASSWORD":WAIT
780 "C1" FOR Z=0 TO 3
790 "PZ"PZ$(Z)=INKEY$:IF PZ$(Z)="" THEN "PZ"
800 WAIT 10:PRINT LEFT$("****",Z+1):WAIT:NEXT Z
810 CH=1:FOR Z=0 TO 3:IF PZ$(Z)<>MID$(PWS(0),Z+1,1)
    LET CH=0
820 NEXT Z:IF CH=1 RETURN
830 WAIT 64:PRINT "TRY AGAIN ...":WAIT:GOTO "C1"
840 "M" PRINT #KONTO";MP,MW
850 PRINT #NP,NW,IW,LE,IN$,W$(*),P$(*),EX(*),NE(*),
    SY(*),PO(*),PWS(*),PA,SO(*),HA(*)
860 END

```

```

870 " " GOSUB "CH":IF ZZ=1 INPUT "DELETE OLD Y/N?";IN$:
    IF IN$<>"Y" END
880 CLEAR:MP=0:MW=0:NP=0:NW=0:IW=0:LE=0:IN$="":PA=0
890 INPUT #"KONTO";MP,MW:GOSUB "IL"
900 INPUT #NP,NW,IW,LE,IN$,W$(*),P$(*),EX(*),NE(*),
    SY(*),PO(*),PW$(*),PA,SO(*),HA(*)
910 BEEP 2:END
920 "IL" DIM SO(MW,MP),HA(MW,MP),W$(MW)*2,P$(MP)*1,
    EX(MW),NE(MP),SY(MP),PO(MP)
930 DIM PW$(0)*4,PSS$(0)*1,WSS$(0)*2,TES(0)*1,T2$(0)*1,
    PZ$(3)*1:ZZ=1:RETURN
940 "C" GOSUB "CH":PRINT USING;IN$+" "+W$(IW):END

```

Seenavigation

Beschreibung:

Das Programm ist geschrieben für einen Sharp PC-1401 mit 10kB. Es dient zur rechnerunterstützten Navigation auf See. Es wurde sorgfältig erstellt und bereits auf mehreren Atlantikfahrten getestet und verwendet. Es kann allerdings keine Haftung für aus der Benutzung des Programms entstandene Navigationsfehler übernommen werden. Wie immer gilt: sei dir nie zu sicher!

Das Programm gliedert sich hauptsächlich in zwei Teile: in einen Teil zur Astronavigation und einen Teil zur Koppelnavigation. Mit dem Teil Astronavigation läßt sich eine Standlinie aus einer Sonnenbeobachtung ermitteln. Dazu werden zwei Verfahren angeboten («DEF»«H» und «DEF»«C»). Die Mittagszeit kann vorausberechnet werden. Auch die Mittagsbreite läßt sich berechnen. Ebenfalls möglich ist eine Standlinie aus der Beobachtung eines Fixsterns, sowie eine Vorausberechnung des LHA. Die Nummern der insgesamt 57 Fixsterne richten sich nach dem englischen Nautical Almanac.

Alle benötigten Daten dazu sind im Programm enthalten. Das Programm ist gültig für das Jahr 1987. Für die folgenden Jahre müssen die Daten im ersten Programmteil jeweils erneuert werden.

Starten des Programms :

Da das Programm den kompletten Speicherplatz benötigt, sollten bei Verwendung eines 10 kB-Rechners die beiden ersten Zeilen gelöscht werden. Sonst könnte für Berechnungen zu wenig Platz sein. Nach dem Laden wird das Programm beim ersten Mal mit der Tastenfolge «SHIFT» «RUN» «ENTER» gestartet. Damit werden die Daten eingelesen. Der weitere Programmablauf ist menügesteuert und erklärt sich von selbst. Im Zweifel immer die «ENTER» Taste betätigen.

Zu Beginn erscheint der Programmname und die Versionsnummer. Dann wird nach (DEF) TASTE ? gefragt. Hiermit sind die definierbaren Tasten des Rechners gemeint. Es muß also die Tastenfolge «DEF» «Taste» gedrückt werden. Da anfangs die Tastenbelegung unbekannt ist, einfach «ENTER» drücken. Das Menü erscheint dann fortlaufend. Mit den definierbaren Tasten kommt man jederzeit in den gewünschten Programmteil. Mit «DEF» «S» kommt man zum Beginn des Hauptmenüs.«DEF»«A» und «DEF»«K» zeigen die jeweiligen Untermenüs. Mit «DEF» «SPC» wird das Programm beendet.

Bedienung:

Wird während des Programmablaufs nach JA oder NEIN gefragt, muß lediglich «J» oder «N» gedrückt werden. Die Anzeige z.B. einer Position erfolgt in zwei Schritten. Dazwischen muß «ENTER» gedrückt werden. Möchte man das vorherige Ergebnis noch einmal sehen, betätigt man die Taste «Pfeil nach oben», danach wieder «ENTER». Mit einem weiteren «ENTER» kommt man nach Anzeige des Ergebnisses wieder zum Hauptmenü.

Die Eingabeformate müssen folgende Form haben:

Uhrzeit	: hh.mmss		hh - Stunde	mm - Minuten
Datum	: dd.MM		dd - Tag	MM Monat
Breite/Länge	: gg.mmss		gg - Grad	ss - Sekunden
Nord/Ost (N/E)	sind positiv (+)			
Süd/West (S/W)	sind negativ (-)			

Belegung der definierbaren Tasten «DEF» «Taste»:

S - Start des Programms
 Spc - Ende des Programms

A - Astronavigation
 B - Mittagsbreite
 C - Line of Position (Standlinie aus Lat/Lon Angaben)
 F - Fixsterne, Vorausberechnung und Standlinie
 H - Standlinie aus Sonnenhöhe
 Z - Mittagszeit

K - Koppelnavigation
 D - Distanz und Kurs zum Ziel
 G - Großkreisberechnung, Kurs und Distanz
 L - Luvgewinn
 N - Neuer Ort
 V - Vektoraddition (Strom)

Listing:

```

5  REM JAHR 1987
10 FE=28 :REM TAGE DES FEBRUARS
20 DATA 01.01,179.21,359.881,.0009,-23.06,.077,.0037
   ,11.01,178.1,359.889
21 DATA .0014,-21.92,.152,.0033,21.01,177.23,359.928,
   .0016,-20.07,.219,.0029
22 DATA 31.01,176.67,359.961,.0016,-17.59,.277,.0023
   ,10.02,176.44,359.995
23 DATA .0015,-14.59,.322,.0018,20.02,176.54,360.026,
   .0012,-11.19,.358,.0012
24 DATA 02.03,176.91,360.050,.0008,-7.49,.382,.0006
   ,12.03,177.49,360.067
25 DATA .0004,-3.61,.393,.0001,22.03,178.2,360.076,
   .0,.33,.396,-.0004
26 DATA 01.04,178.96,360.075,-.0004,4.25,.388,-.0009
   ,11.04,179.67,360.068
27 DATA -.0007,8.04,.37,-.0014,21.04,180.28,360.052,
   -.001,11.6,.343,-.0018
28 DATA 01.05,180.7,360.031,-.0011,14.85,.307,-.0023
   ,11.05,180.9,360.009

```

```

29 DATA -.0011,17.69,.26,-.0026,21.05,180.88,359.984,
  -.001,20.03,.208,-.003
30 DATA 31.05,180.62,359.964,-.0006,21.81,.148,-.0033
  ,10.06,180.2,359.951
31 DATA -.0003,22.96,.081,-.0034,20.06,179.68,359.944
  ,.0002,23.43,.013,-.0034
32 DATA 30.06,179.14,359.949,.0007,23.22,-.056,-.0033
  ,10.07,178.7,359.963
33 DATA .001,22.33,-.122,-.003,20.07,178.43,359.984,
  .0012,20.81,-.183,-.0028
34 DATA 30.07,178.39,360.009,.0012,18.7,-.24,-.0023,
  09.08,178.6,360.034,.0011
35 DATA 16.07,-.286,-.0019,19.08,179.05,360.056,.0009
  ,13.02,-.324,-.0015
36 DATA 29.08,179.7,360.074,.0006,9.63,-.355,-.001,
  08.09,180.5,360.087,.0002 37 DATA 5.98,-.375,-.0006,18.09,18
38 DATA 28.09,182.26,360.085,-.0006,-1.72,-.39,.0003,
  08.10,183.05,360.073
39 DATA -.0011,-5.59,-.284,.0009,18.10,183.67,360.05,
  -.0014,-9.34,-.366,.0014
40 DATA 28.10,184.03,360.022,-.0016,-12.86,-.338,
  .0019,07.11,184.09,359.988
41 DATA -.0017,-16.05,-.3,.0025,17.11,183.8,359.953,
  -.0016,-18.8,-.249,.003
42 DATA 27.11,183.17,359.92,-.0013,-20.99,-.188,
  .0035,07.12,182.24,359.893
43 DATA -.0008,-22.52,-.119,.0038,17.12,181.09,
  359.877,-.0001,-23.33,-.042
44 DATA .0039,27.12,179.85,359.876,.0006,-23.36,
  .037,.0038,0,0,0,0,0,0
60 "NJ":DATA 358.07,29.02,353.58,-42.37,350.06,56.47,
  349.26,-18.06,335.88,-57.31
70 DATA 328.38,23.41,315.55,-40.37,314.59,4.04,309.14
  ,49.82,291.20,16.49
80 DATA 281.51,-8.21,281.06,45.99,278.88,6.34,278.62,
  28.60,276.10,-1.21
90 DATA 271.37,7.41,264.08,-52.69,258.85,-16.70,
  255.46,-28.95,245.34,5.26
100 DATA 243.86,28.06,234.45,-59.46,223.12,-43.38,
  221.74,-69.66,218.26,-8.60
110 DATA 208.07,12.03,194.26,61.82,182.89,14.64,176.21
  ,-17.47,173.56,-63.02
120 DATA 172.42,57.03,166.62,56.03,158.85,-11.10,
  153.24,49.37,149.31,-60.31
130 DATA 148.52,-36.31,146.23,19.24,140.35,-60.46,
  137.46,-15.99,137.31,74.20
140 DATA 126.46,26.76,112.84,-26.41,108.23,-69.0,
  102.59,-15.71,96.81,-37.1
150 DATA 96.42,12.57,90.92,51.49,84.17,-34.39,80.87,
  38.77,76.38,-26.32
160 DATA 62.46,8.83,53.88,-56.71,49.75,45.23,34.11,
  9.82,28.14,-47.02
170 DATA 15.76,-29.69,13.97,15.14
180 DATA 0,31,59,90,120,151,181,212,243,273,304,334,365
580 DIM FA(57):DIM FD(57):DIM DY(12)
590 RESTORE "NJ":FOR I=1 TO 57:READ FA(I):
  READ FD(I):NEXT I
600 FOR I=0 TO 12:READ DY(I):NEXT I
610 IF FE=29 THEN FOR I=2 TO 12:DY(I)=DY(I)+1:NEXT I

```

```

620 RESTORE:GOTO "S"
630 "ARIES":GH=99.13+360.98565*T:RETURN
640 "S":PAUSE "* S E A N A V *":PAUSE "VERS.2.9**1.9.87"
650 WAIT:PRINT "(DEF) TASTE ? "
660 WAIT 90:PRINT "ASTRO-NAV   = A":
    PRINT "KOPPEL-NAV   = K"
670 WAIT 0:PRINT "TASTE A/K ? ":CALL 5151:
    P=PEEK 24634:IF P=75 GOTO "K"
680 "A":WAIT 90:PRINT "== ASTRO-NAV ==":
    PRINT "MITTAGS-ZEIT = Z"
690 PRINT "MITGS-BREITE = B":PRINT "AZIMUT/DH   = H"
700 PRINT "LOP           = C":PRINT "FIXSTERNE   = F"
710 WAIT:PRINT "KEY Z/B/H/C/F ?":GOTO "A"
720 "K":WAIT 90:PRINT "== KOPPEL-NAV ==":
    PRINT "ZIELDST/-KURS =D"
730 PRINT "NEUER ORT     = N":PRINT "LUVGEWINN   = L"
740 PRINT "VEKTORADD    = V":PRINT "GROSSKREIS   = G"
750 WAIT:PRINT "KEY D/N/L/V/G ?":GOTO "K"
760 "Z":PAUSE "   MITTAGSZEIT"
780 GOSUB "DATUM":GOSUB "MITTZEIT":G=MI:Q=1
790 GOSUB "AUSGABE":WAIT:PRINT R$;S$:GOTO "S"
800 "B":PAUSE "   MITTAGSBREITE"
810 GOSUB "HOEHE":GOSUB "MITTZEIT":T=TA+MI/24:
    GOSUB "DEC"
820 G=K+90-HW:Q=5:GOSUB "AUSGABE":WAIT:
    PRINT "LAT   ";R$;S$:GOTO "S"
970 "H":PAUSE "   AZIMUT / DH":GOSUB "EINGABE"
990 HC=ASN(SIN DC*SIN B1+COS DC*COS B1*COS(GH+L1))
1000 DH=(HW-HC)*60
1010 AZ=ACS((SIN DC-SIN B1*SIN HC)/(COS HC*COS B1))
1020 IF SIN(GH+L1)<0 GOTO 1040
1030 AZ=360-AZ
1040 WAIT:IF ABS(DH)>=1E3 THEN PRINT "FALSCHE EINGABE":
    GOTO "S"
1050 WAIT:PRINT USING "####.#";"DELTAH=";DH;" SM"
1060 WAIT 0:PRINT USING "####.#";"AZIMUT=";AZ
1062 CALL 5151:P=PEEK 24634:IF P=4 GOTO 1050
1065 GOTO "S"
1070 "C":PAUSE "   == LOP =="
1090 GOSUB "EINGABE":CA=SGN(180-GH+L1)
1110 CL=CA*ACS((SIN HW-SIN DC*SIN B1)/COS DC/COS B1)-GH
1120 CF=SIN CL/ABS SIN CL
1130 CL=COS CL:CL=ACS CL:CL=CF*CL
1140 G=CL:Q=6:GOSUB "AUSGABE"
1150 WAIT:PRINT "LON   ";R$;S$
1160 WAIT 0:PRINT "WEITER?(J/N) "
1170 CALL 5151:P=PEEK 24634:IF P=78 GOTO "S"
1180 INPUT "BREITE ? ";BA:B1=DEG BA:GOTO 1110
1190 "F":PAUSE "   FIXSTERNE"
1200 WAIT 0:PRINT "VORAUSBER?(J/N)"
1210 CALL 5151:P=PEEK 24634:IF P=74 GOTO 1240
1220 INPUT "WELCHER STERN? ";F
1230 LX=1:GOSUB "HOEHE"
1240 INPUT "DATUM(DD.MM): ";DT
1250 MO=(DT-INT(DT))*100:TA=INT(DT)+DY(MO-1)
1270 INPUT "ZEIT ? ";ZE
1290 T=TA+DEG ZE/24:IF P=74 GOTO 1340
1300 INPUT "BREITE ? ";BA:B1=DEG BA
1310 INPUT "LAENGE ? ";LA:L1=DEG LA

```



```

1320 GOSUB "ARIES":GH=GH+FA(F)
1330 GH=GH-INT(GH/360)*360:DC=FD(F):GOTO 990
1340 INPUT "LAENGE ? ";LA:L1=DEG LA
1350 GOSUB "ARIES":GH=GH-INT(GH/360)*360-L1
1370 WAIT:PRINT USING "*****.#";"LHA =";GH;" GRAD"
1380 P=0:GOTO 1220
1390 "D":PAUSE " ZIELDIST/-KURS"
1400 GOSUB "ORTE":DB=(B1-B2)*60:DL=(L1-L2)*60
1420 IF(DB=0)AND(L2<L1)LET K=270:GOTO 1560
1430 IF(DB=0)AND(L2>L1)LET K=90:GOTO 1560
1440 IF(L1=L2)AND(B2<B1)LET K=180:GOTO 1530
1450 VB=(180*60/PI)*(LN(TAN(45+B2/2))-LN(TAN(45+B1/2)))
1460 K=ATN(ABS(DL)/VB)
1470 IF(L1<L2)AND(B1>B2)LET K=180-ABS(K)
1480 IF(L1>L2)AND(B1<B2)LET K=360-K
1530 DI=ABS(DB/COS K):GOTO 1570
1560 DI=ABS(DL*COS B1)
1570 GOSUB "NORM"
1580 WAIT:PRINT USING "*****.#";"DIST=";DI;" SM"
1590 WAIT 0:PRINT USING "*****.#";"KURS=";K;" GRAD"
1595 CALL 5151:P=PEEK 24634:IF P=4 GOTO 1580
1600 GOTO "S"
1640 "N":PAUSE " NEUER ORT":LX=0
1660 INPUT "LAT A ? ";BA
1670 INPUT "LON A ? ";LA
1680 GOSUB "RECKON"
1750 SM=SM/60:Y=REC(SM,TH):B1=DEG BA:L1=DEG LA
1770 DB=Z:B2=B1+DB:DL=Y/COS(B1+DB/2):L2=L1+DL
1790 IF LX=1 GOTO 1920
1800 G=B2:Q=5:GOSUB "AUSGABE"
1810 WAIT:PRINT "LAT ";R$;S$
1820 G=L2:Q=6:GOSUB "AUSGABE"
1830 PRINT "LON ";R$;S$:GOTO "S"
1870 "L":PAUSE " LUVGEWINN"
1880 GOSUB "ORTE":GOSUB "GROSSKREIS"
1900 GE=GD:B3=B2:L3=L2:LX=1:GOTO 1680
1920 B1=B2:L1=L2:B2=B3:L2=L3
1930 GOSUB "GROSSKREIS":LV=GE-GD
1950 WAIT:PRINT USING "*****.#";"LUVGEWINN ";LV;" SM"
1960 WAIT 0:PRINT USING "*****.#";"KURS ";GK;" GRAD"
1965 CALL 5151:P=PEEK 24634:IF P=4 GOTO 1950
1970 GOTO "S"
1980 "G":PAUSE " GROSSKREIS"
1990 GOSUB "ORTE":GOSUB "GROSSKREIS"
2010 WAIT:PRINT USING "*****.#";"DIST=";GD;" SM"
2020 WAIT 0:PRINT USING "*****.#";"KURS=";GK;" GRAD"
2025 CALL 5151:P=PEEK 24634:IF P=4 GOTO 2010
2030 GOTO "S"
2120 "V":PAUSE " VEKTORADDITION":VB=0:VL=0
2130 GOSUB "RECKON"
2140 Y=REC(SM,TH):VB=VB+Z:VL=VL+Y
2170 WAIT 0:PRINT "WEITER?(J/N) ":CALL 5151:
P=PEEK 24634:IF P=78 GOTO 2200
2190 GOTO 2130
2200 Y=POL(VB,VL)
2210 IF Z>=0 GOTO 2230
2220 Z=360+Z
2230 USING:WAIT:PRINT "DISTANZ=";Y
2240 PRINT "RICHTUNG=";Z:GOTO "S"

```

```

2250 "ORTE":INPUT "LAT A (S=-)? ";BA
2260 INPUT "LON A (W=-)? ";LA
2270 INPUT "LAT B (S=-)? ";BB
2280 INPUT "LON B (W=-)? ";LB
2290 B1=DEG BA:L1=DEG LA:B2=DEG BB:L2=DEG LB:RETURN
2300 "RECKON":INPUT "DISTANZ ? ";SM
2310 INPUT "RICHTUNG ? ";K:TH=K
2320 IF(K>=0)AND(K<=270)LET TH=90-K:RETURN
2330 IF(K>270)AND(K<=360)LET TH=450-K:RETURN
2340 RETURN
2350 "GROSSKREIS":GD=ACS(SIN B1*SIN B2+COS B1*
COS B2*COS(L1-L2))
2360 GK=ACS((SIN B2-COS GD*SIN B1)/(SIN GD*COS B1))
2370 IF L1<L2 GOTO 1920
2380 GK=360-GK:GD=GD*60:RETURN
2400 "NORM":IF K<0 LET K=ABS(K)+180
2420 IF K<=360 THEN RETURN
2430 K=K-INT(K/360)*360:RETURN
2480 "EINGABE":GOSUB "HOEHE":GOSUB "DATUM"
2510 INPUT "ZEIT ? ";ZE
2520 T=TA+DEG ZE/24
2530 INPUT "BREITE ? ";BA:B1=DEG BA
2540 INPUT "LAENGE ? ";LA:L1=DEG LA
2550 GOSUB "GHA":GH=K:GOSUB "DEC":DC=K:RETURN
2600 "AUSGABE":GG=DMS(ABS(G)):DZ=GG-INT(GG):GG=INT(GG)
2620 MM=INT(DZ*100):SS=100*(DZ*100-MM):DZ=SS-INT(SS)
2650 IF DZ<0.5 GOTO 2670
2660 SS=SS+1
2670 IF(SS>=60) LET SS=0:MM=MM+1
2680 SS=INT(SS)
2710 QU$="":USING "###.#"
2720 IF(Q=5)AND(G=0)GOTO 2840
2730 IF(Q=6)AND(G=0)GOTO 2840
2740 IF(Q=5)AND(G>0)LET QU$=" N"
2750 IF(Q=5)AND(G<0)LET QU$=" S"
2760 IF(Q=6)AND(G>0)LET QU$=" E"
2770 IF(Q=6)AND(G<0)LET QU$=" W"
2780 IF(Q=1)LET QU$=" UTC"
2840 N$="":O$=""
2850 IF MM>=10 GOTO 2870
2860 N$="0"
2870 IF SS>=10 GOTO 2900
2880 O$="0"
2900 GG$=STR$ GG:MM$=STR$ MM:SS$=STR$ SS
2910 R$=GG$+" "+N$+MM$+" ":S$=O$+SS$+QU$:RETURN
2940 "DATUM":INPUT "DATUM(DD.MM): ";DT
2960 MO=(DT-INT(DT))*100:M1=MO-M:DM=0
2980 ON (M1+1) GOTO 3010,3000
2990 GOTO 3030
3000 GOSUB "MONATE"
3010 TA=INT(DT)-DD+DM
3020 IF TA<=10 THEN RETURN
3030 READ DA,H,A,B,E,C,D
3040 IF DA=0 LET M=0:GOTO 3100
3050 DD=INT(DA):M=(DA-INT(DA))*100:TA=INT(DT)-DD
3080 GOTO 2960
3100 RESTORE :WAIT:PRINT "NEUE EINGABEN !":GOTO "S"
3120 "MONATE":IF MO=3 LET DM=FE:RETURN
3210 IF(MO=5)OR(MO=7)OR(MO=10)OR(MO=12)LET DM=30:RETURN

```

```

3220 DM=31:RETURN
3280 "GHA":K=H+A*T+B*SQU T:GOSUB "NORM":RETURN
3320 "DEC":K=E+C*T+D*SQU T:GOSUB "NORM":RETURN
3360 "MITTZEIT":INPUT "GEGIS. LAENGE ";LE:LG=DEG LE
3390 T=TA:GOSUB "GHA":MI=(360-K-LG)/15:RETURN
3420 "HOEHE":INPUT "HOEHE ? ";HS
3440 WAIT 40:PRINT "INDEXFEHLER ";IB
3450 INPUT IB:IC=IB/60
3460 HW=DEG HS
3470 IF LX=1 GOTO 3540
3480 IF HW<=20 GOTO 3600
3490 IF HW<=25 LET GB=.11
3500 IF HW<=40 LET GB=.12
3510 IF HW>40 LET GB=.13
3520 GOTO 3620
3540 LX=0:IF HW<=22 GOTO 3600
3550 IF HW<=35 LET GB=-.05
3560 IF HW<=70 LET GB=-.04
3570 IF HW>70 LET GB=-.03
3580 GOTO 3620
3600 PAUSE "HW BERICHTIGEN"
3610 INPUT "BERICHTIGUNG=" ;GB:GB=GB/100
3620 HW=HW+DEG GB+IC:RETURN
3640 " ":PAUSE "*** E N D E ***":END

```

Berechnung der Molmasse

Beschreibung:

unerlässlich für jeden Chemiker.....

Das einfache Programm berechnet sofort und zuverlässig die Molekularmasse einer beliebigen chemischen Summenformel. Dabei wird jedes Symbol der Summenformel als Element erkannt und dessen Atommasse mit dem nachfolgenden Index multipliziert.

Die Werte der Atommassen wurden aus einem neueren Periodensystem entnommen. Es enthält sämtliche Werte bis zum Element AC

BEDIENUNG:

Nach dem Programmaufruf muß man die gewünschte Summenformel eingeben. Hierbei muß man nach jedem Elementsymbol den Index angeben (auch 1) Es dürfen nur Grossbuchstaben und Zahlen vorkommen. Einzelne Elementsymbole dürfen mehrfach vorkommen. Klammern, Punkte und Symbole dürfen nicht verwendet werden.

Ausgegeben wird Molmasse als Variable M, die man in weitere Berechnungen mit einbeziehen darf. Außerdem wird Variable A als diejenige Zahl gespeichert, die vor dem Programmaufruf im Display stand.

Listing:

```

10 "A":AREAD A
20 REM MOLEKULARMASSE
30 REM HEID DANIEL
40 REM NEUE-ALLMENDSTR
50 REM CH-8703 ERLNBACH

```

```
60 CLS :Y=0:M=0:ERASE Z$( ):ZAS="":ES=""
70 PRINT "*** MOLMASSE ***"
80 PRINT "Bsp ZN1S1O4"
90 INPUT "SUMMENFORMEL? ";SS
100 L=LEN (SS):DIM Z$(L)
110 Y=Y+1
120 Z$(Y)=MID$( SS,Y,1)
130 X=ASC (Z$(Y))
140 IF X>64THEN 160
150 IF X<64THEN 180
160 ES=ES+Z$(Y)
170 GOTO 110
180 ZAS=ZAS+Z$(Y)
190 IF ASC MID$( SS,Y+1,1)>64OR Y=LTHEN 210
200 GOTO 110
210 RESTORE 220
220 FOR Z=1TO 75
230 READ F$,F
240 IF F$=E$THEN 290
250 NEXT Z
260 PRINT "ELEMENT ";E$
270 PRINT "NICHT VORHANDEN"
280 PRINT "A = ";A:END
290 ZA=VAL (ZAS)
300 M=M+ZA*F
310 ZAS="":ES=""
320 IF Y<>LTHEN 110
330 CLS :PRINT "SUMMENFORMEL:"
340 PRINT " M = ";SS;" = ";M
350 PRINT " A = ";A
360 END
370 DATA "H",1.008
371 DATA "HE",4.003
372 DATA "LI",6.941
373 DATA "BE",9.012
374 DATA "B",10.81
375 DATA "C",12.011
376 DATA "N",14.007
377 DATA "O",16.000
378 DATA "F",19.998
379 DATA "NE",20.179
380 DATA "NA",22.990
381 DATA "MG",24.305
382 DATA "AL",26.982
383 DATA "SI",28.085
384 DATA "P",30.974
385 DATA "S",32.06
386 DATA "CL",35.453
387 DATA "AR",39.948
388 DATA "K",39.098
389 DATA "CA",40.08
390 DATA "SC",44.956
391 DATA "TI",47.9
392 DATA "V",50.941
393 DATA "CR",51.996
394 DATA "MN",54.938
395 DATA "FE",55.847
396 DATA "CO",58.933
397 DATA "NI",58.70
398 DATA "CU",63.546
```

```
399 DATA "ZN",65.38
400 DATA "GA",69.72
401 DATA "GE",72.59
402 DATA "AS",74.922
403 DATA "SE",78.96
404 DATA "BR",79.904
405 DATA "KR",83.80
406 DATA "RB",85.468
407 DATA "SR",87.62
408 DATA "Y",88.906
409 DATA "ZR",91.22
410 DATA "NB",92.906
411 DATA "MO",95.94
412 DATA "TC",98
413 DATA "RU",101.07
414 DATA "RH",102.905
415 DATA "PD",106.4
416 DATA "AG",107.868
417 DATA "CD",112.41
418 DATA "IN",114.82
419 DATA "SN",118.69
420 DATA "SB",121.75
421 DATA "TE",127.60
422 DATA "J",126.904
423 DATA "XE",131.30
424 DATA "CS",132.905
425 DATA "BA",137.33
426 DATA "LA",138.905
427 DATA "HF",178.49
428 DATA "TA",180.948
429 DATA "W",183.85
430 DATA "RE",186.207
431 DATA "OS",190.2
432 DATA "IR",192.22
433 DATA "PT",195.09
434 DATA "AU",196.966
435 DATA "HG",200.59
436 DATA "TI",204.37
437 DATA "PB",207.2
438 DATA "BI",208.980
439 DATA "PO",209
440 DATA "AT",210
441 DATA "RN",222
442 DATA "FR",223
443 DATA "RA",226.025
444 DATA "AC",227.028
450 END
```

Adress-Datei

Beschreibung:

Dieses Programm wurde auf einem SHARP PC-1403 mit 8 KB unter Verwendung von TRANSFILE ST 3.0 geschrieben.

1.0 Programmstart:

Starten Sie die Adressdatei nur zum ersten Mal mit "RUN", sonst immer nur mit "DEF" "A" !

ACHTUNG: Wenn Sie die Adressdatei mit "RUN" starten, werden alle Variablen gelöscht, d.h. alle Daten gehen verloren !!!

2.0 Hauptmenü:

Nach dem Programmstart erscheint zunächst im Display:

```
' *** Adressdatei *** '
```

Dann folgt nach kurzer Initialisierung das Hauptmenü:

```
'A,E,S,L,V,Z? - '
```

Hier sind Sie durch Eingabe des entsprechenden Buchstabens in der Lage, die gewünschte Funktion aufzurufen:

A ... Abbruch	- Programmende
E ... Eingabe	- Adresse eingeben
S ... Suchen	- Adresse suchen
L ... Löschen	- Adresse löschen
V ... Von TRANSFILE	- Adressen von TRANSFILE in den SHARP übertragen
Z ... Zu TRANSFILE	- Adressen von SHARP auf TRANSFILE übertragen

Im folgenden werden die einzelnen Funktionen beschrieben.

2.1 Eingabe

Diese Funktion wird mittels "E" "ENTER" vom Hauptmenü aus aufgerufen. Geben Sie hier bitte die gewünschten Daten gefolgt von "ENTER" in Ihren SHARP PC ein. Um eine Eingabe zu überspringen, geben Sie nur "ENTER" ein. Dabei muß mindestens entweder der Vor- oder der Nachname eingegeben werden, sonst erscheint eine Fehlermeldung.

2.2 Suchen

Nach dem Anwählen dieser Funktion mit "S" "ENTER" kommt es zur folgenden Anzeige:

```
'SUCHE NACH: - '
```

Antworten Sie hierauf mit dem zu suchenden Namen. Dabei wird der Vor- und der Nachname auf den eingegebenen Namen hin untersucht. Ein Beispiel soll dies verdeutlichen:

Sie geben folgendes in Ihren SHARP ein:

```
'SUCHE NACH: MA- '
```

Nun findet Ihr Pocketcomputer zum Beispiel "MARKUS SCHMIDT". Die Adresse können Sie jetzt mittels "ENTER" durchscrollen. Danach erscheint folgende Abfrage auf dem Display:

```
'Abbruch, Weiter? - '
```

Mit "A" "ENTER" (für Abbruch) gelangen Sie wieder ins Hauptmenü. Tippen Sie jedoch "W" "ENTER" (für Weiter) ein, so wird in unserem Beispiel nach dem nächsten Vorkommen von "MA" gesucht. Nun findet Ihr SHARP zum Beispiel "EMIL MAIER". Jetzt beginnt alles wieder von vorne, d.h., Sie können nun die Adresse wieder mit "ENTER" durchscrollen, usw. Wird der SHARP PC nicht mehr fündig, so kehrt er ins Hauptmenü zurück.

2.3 Löschen

Nach dem Anwählen dieser Funktion mit "E" "ENTER" kommt es zur folgenden Anzeige:

```
'LOESCHEN: -
```

Antworten Sie hierauf mit dem zu löschenden Namen. Dabei werden die Adressen gemäß Punkt "2.2 Suchen" durchsucht. Es erfolgt lediglich eine etwas andere Abfrage:

```
'LOESCHEN (J,W,A)? -
```

Mit "A" "ENTER" (für Abbruch) gelangen Sie wieder ins Hauptmenü, mit "W" "ENTER" (für Weiter) wird nach dem nächsten Vorkommen gesucht (vgl. 2.2 Suchen).

Tippen Sie jedoch "J" "ENTER" (für Ja), so wird die gesamte Adresse aus dem Speicher des SHARP PCs gelöscht! Dies kann einige Zeit dauern, da die ganzen Feldvariablen der Adressen umformatiert werden müssen.

2.4 Von TRANSFILE

Durch Drücken von "V" "ENTER" im Hauptmenü, werden die Adressen als Variablen von TRANSFILE ST in den SHARP eingelesen. Es erscheinen nun folgende Zeilen auf dem Display Ihres SHARPs:

```
'DATEN VOM ATARI LADEN
'ATARI VORBEREITEN !!!
'»ENTER« FUER START !!
```

Laden Sie dazu eine zuvor auf Diskette gespeicherte Datei mit den zu übertragenden Adressen in den Speicher von TRANSFILE ST. (Die Datei mit den Variablen muß binär auf Diskette vorliegen! Siehe TRANSFILE Handbuch!)

Bereiten Sie Ihren ATARI auf die Übertragung zum SHARP vor, drücken Sie am SHARP "ENTER", und starten Sie zuletzt die Übertragung am ATARI.

Nach einiger Zeit ist die Übertragung beendet und der SHARP kehrt ins Hauptmenü zurück.

2.5 Zu TRANSFILE

Mit "Z" "ENTER" können Sie die Adressen als Variablen an TRANSFILE senden.

Es erscheinen nun folgende Zeilen auf dem Display Ihres SHARPs:

```
'DATEN ZUM ATARI SENDEN
'ATARI VORBEREITEN !!!
'»ENTER« FUER START !!
```

Bereiten Sie Ihren ATARI auf die Übertragung vom SHARP vor, und drücken Sie am SHARP "ENTER" um die Übertragung zu starten. Nach einiger Zeit ist die Übertragung beendet, und der SHARP kehrt ins Hauptmenü zurück. Speichern Sie nun die übertragenen Variablen am ATARI mittels dem TRANSFILE-Menüpunkt 'Speichern' binär auf Diskette ab.

3.0 Zum Programm selbst

In Zeile 160 steht folgendes:

```
160 MA=50
```

Die Variable MA gibt die maximale Adressenanzahl an. Diesen Wert können Sie selbstverständlich dem freien Speicher Ihres SHARPs anpassen.

Jedoch vorsicht beim Übertragen von Adressen von TRANSFILE zum SHARP !!!

Hier kann es zu einem Übertragungsfehler kommen, wenn Sie versuchen, Adressen einzulesen, die vorher mit einem anderen Maximalwert abgespeichert wurden.

Desweiteren läßt sich in Zeile 170 die jeweils maximale Anzahl der Buchstaben bzw. Ziffern, die man pro Adresseintrag eingeben kann, einstellen. Im vorliegenden Programm ist diese Anzahl wie folgt eingestellt:

```
Vorname : 10 Buchstaben
Nachname : 10 Buchstaben
Straße : 15 Buchstaben
Wohnort : 20 Buchstaben
Telefon : 12 Buchstaben
```

Listing:

```
100 REM *****
110 REM ** ADRESSDATEI **
120 REM ** FUER SHARP 1403 **
150 REM *****
160 MA=50
170 DIM VN$(MA)*10, NN$(MA)*10, ST$(MA)*15, WO$(MA)*20,
    TE$(MA)*12
180 "A"
190 PAUSE " *** Adressdatei ***"
200 E$=""
210 INPUT "A,E,S,L,V,Z? "; E$: E$=LEFT$(E$, 1)
220 IF E$="A" THEN END
230 IF E$="E" THEN GOSUB 300
240 IF E$="S" THEN LET F=0:GOSUB 410
250 IF E$="L" THEN LET F=1:GOSUB 410
260 IF E$="V" THEN GOSUB 730
270 IF E$="Z" THEN GOSUB 810
280 GOTO 200
290 REM *** EINGABE ***
300 NR=NR+1:G=0
310 IF NR>MA THEN BEEP 3:PAUSE " *** SPEICHER VOLL !!! ":
    NR=NR-1:RETURN
320 INPUT "VORNAME: "; VN$(NR)
330 INPUT "NACHNAME: "; NN$(NR)
340 IF VN$(NR)="" AND NN$(NR)="" THEN LET G=1
```



```
350 IF G=1 THEN BEEP 1:PAUSE " *** EINGABEFehler !!!":
NR=NR-1:G=0:RETURN
360 INPUT "STRASSE: ";ST$(NR)
370 INPUT "WOHNORT: ";WO$(NR)
380 INPUT "TELEFON: ";TE$(NR)
390 RETURN
400 REM *** SUCHE ***
410 H=0
420 IF F=0 THEN INPUT "SUCHE NACH: ";S$
430 REM *** LOESCHEN ***
440 IF F=1 THEN INPUT "LOESCHEN: ";S$
450 FOR I=1 TO NR
460 IF LEFT$(VN$(I),LEN(S$))=S$ OR LEFT$(NN$(I),
LEN(S$))=S$ THEN GOSUB 510
470 IF H=1 THEN LET I=NR:H=0
480 NEXT I
490 RETURN
500 REM *** AUSGABE AUF DISPLAY ***
510 IF VN$(I)<>" " THEN PRINT VN$(I)
520 IF NN$(I)<>" " THEN PRINT NN$(I)
530 IF ST$(I)<>" " THEN PRINT ST$(I)
540 IF WO$(I)<>" " THEN PRINT WO$(I)
550 IF TE$(I)<>" " THEN PRINT TE$(I)
560 D$=""
570 IF F=0 THEN INPUT "Abbruch,Weiter? ";D$
580 IF F=1 THEN INPUT "LOESCHEN (J,W,A)? ";D$
590 D$=LEFT$(D$,1)
600 IF D$="A" THEN LET H=1:RETURN
610 IF D$="W" THEN RETURN
620 IF D$="J" AND F=1 THEN GOSUB 650:RETURN
630 GOTO 560
640 REM *** ADRESSE LOESCHEN
650 FOR J=I TO NR-1
660 VN$(J)=VN$(J+1):NN$(J)=NN$(J+1):ST$(J)=ST$(J+1)
670 WO$(J)=WO$(J+1):TE$(J)=TE$(J+1)
680 NEXT J
690 VN$(NR)="" : NN$(NR)="" : ST$(NR)="" : WO$(NR)="" :
TE$(NR)=""
700 NR=NR-1:I=I-1
710 RETURN
720 REM *** VOM ATARI
730 PAUSE "DATEN VOM ATARI LADEN"
740 PAUSE "ATARI VORBEREITEN !!!"
750 PRINT ">ENTER< FUER START !!"
760 INPUT *VN$(*),NN$(*),ST$(*),WO$(*),TE$(*)
770 NR=VAL(VN$(0))
780 PAUSE " *** FERTIG ***"
790 RETURN
800 REM *** ZUM ATARI
810 PAUSE "DATEN ZUM ATARI SENDEN"
820 PAUSE "ATARI VORBEREITEN !!!"
830 PRINT ">ENTER< FUER START !!"
840 VN$(0)=STR$(NR)
850 PRINT *VN$(*),NN$(*),ST$(*),WO$(*),TE$(*)
860 PAUSE " *** FERTIG ***"
870 RETURN
```

Flugvorbereitung

Beschreibung:

Programme für die Flugvorbereitung

DEF A Entfernung - Kurs

Kennung von Start- und Zielflughafen eingeben. Es wird Entfernung und missweisender Kurs berechnet. Weiterhin Eigengeschwindigkeit (true Airspeed) Windrichtung und Stärke, Vorhaltewinkel (Luvwinkel), der korrigierte Kurs und die korrigierte Geschwindigkeit. Weitere Flugplätze können in Zeile 410 angefügt werden.

DEF D Vor-Navigation

Funkfeuer zur Navigation nutzen. Nach Eingabe von VOR-Koordinaten und deren Kennungen werden Kurs u. Entfernung zum jeweiligen Funkfeuer angezeigt.

DEF K Umrechnung Knoten in Kilometer/h

DEF L Umrechnung Kilometer/h in Knoten

DEF H eine ungenaue Uhr

DEF S Dual in Dezimal

DEF S PC Dezi in Dual

DEF . Lottozahlen

DEF x JAPANISCH für Anfänger

DEF V Text scrollen

DEF N Sortieren von Text u. Zahlen

DEF C Wind erfliegen

DEF B Vorhaltewinkel und Flugzeit

Listing:

1005	P	1085	RS=49130707
1010	PRINT "*** ENTFERNUNG-KURS ***"	1090	DS=48410913
1015	RK=50200732	1095	CM=50500611
1020	TK=48590820	1100	TA=48471016
1025	KS=50430805	1105	FQ=51020841
1030	VF=49360822	1110	KD=51190743
1035	KB=50460710	1115	YA=48161225
1040	GB=50410810	1120	QF=49221040
1045	DW=53030847	1125	YB=49081259
1050	DL=51170645	1130	LA=51290754
1055	DF=50020834	1135	FC=49560904
1060	RZ=49160730	1140	MA=48261056
1065	VV=52280941	1145	RF=49280812
1070	DK=50520709	1150	VA=51511002
1075	DM=48081142	1155	FK=50131004
1080	DN=39301105	1160	TB=48480811

1165	EJ=49551055	1460	GK=51150853
1170	QD=49591138	1465	LK=51230635
1175	YG=49011129	1470	QK=50081128
1180	KF=51030742	1475	RL=49200812
1185	KI=50490750	1480	ML=48311202
1190	MB=48070946	1485	WL=53450730
1195	LI=51580833	1490	WF=53160727
1200	LB=51470717	1495	WD=53090837
1205	WR=53360643	1500	YL=47521001
1210	VE=52191033	1505	KL=51010700
1215	WB=53300834	1510	QL=50091103
1220	QE=49481108	1515	HL=53481043
1225	VC=52411007	1520	HC=53011109
1230	QY=50141100	1525	FU=49420911
1235	KV=50240632	1530	OT=49580809
1240	WC=52290811	1535	FM=49280831
1245	MW=48501253	1540	FN=50530849
1250	RW=50340739	1545	LM=51390710
1255	LD=51370652	1550	KZ=51060736
1260	TD=47580831	1555	TM=48030922
1265	MQ=48421051	1560	KM=51180814
1270	LW=51310737	1565	FO=49410859
1275	FE=49580839	1570	GM=49240908
1280	ME=48241244	1575	MY=48171230
1285	WE=53240714	1580	LT=51570746
1290	LE=51240656	1585	TN=48370929
1295	XF=54470923	1590	SN=48041138
1300	TF=48010750	1595	QN=49171127
1305	TY=47400931	1600	HN=54050957
1310	MF=48311321	1605	YN=49131218
1315	WQ=53020830	1610	WS=53380711
1320	FG=50120910	1615	GN=51140849
1325	LF=51200622	1620	WY=53430714
1330	HB=54151102	1625	WN=52280711
1335	MG=48291017	1630	VN=51421002
1340	MH=49071047	1635	LO=51560840
1345	HI=53320950	1640	WH=53040819
1350	LH=51410749	1645	WO=52170758
1355	HM=53551003	1650	LP=51370837
1360	QT=50011032	1655	VP=52241014
1365	XB=54090854	1660	VY=52130852
1370	XH=54110755	1665	YR=49091205
1375	QH=49351053	1670	FB=50200853
1380	TH=48480956	1675	XR=54130936
1385	UH=52100957	1680	XE=52170730
1390	VH=52460937	1685	VR=52110904
1395	QM=50171151	1690	QP=49521147
1400	VL=5370924	1695	FR=49241043
1405	VI=51480923	1700	RJ=49190640
1410	MI=48141008	1705	VS=52091026
1415	SI=48431132	1710	OP=49070947
1420	MJ=48111108	1715	TX=49080947
1425	WJ=53410704	1720	FS=50011015
1430	LC=51320632	1725	TE=48040834
1435	TK=48590820	1730	US=51350813
1440	VK=51250923	1735	RY=49180827
1445	MK=47421020	1740	LS=52000651
1450	CK=54231009	1745	XM=53590909
1455	TZ=47410908	1750	XO=54190841

```

1755 MT=48011006
1760 RT=49520647
1765 VU=52591028
1770 HI=53390942
1775 WU=52550803
1780 WV=52580917
1785 MP=48261221
1790 MV=48381312
1795 YV=47571212
1800 WG=53470755
1805 QW=49411207
1810 KW=51180749
1815 LX=51400636
1820 WM=53030913
1825 XW=54550821
1830 WI=53300803
1835 KM=51080723
1840 GW=51190911
1845 FV=49360822
1850 FW=49490954
1855 XY=54410832
1860 V6=50180832
1865 V3=49440720
1870 V9=49580729
1875 V4=50340708
1880 V8=50360757
1885 S1=50250749
1890 S2=50270647
1895 S3=50110753
1900 S4=50160751
1905 S5=50300715
1910 M1=50130645
1915 M2=50180719
1920 M3=50130727
1925 M4=50230701
1930 M5=50380627
1935 M6=50130800
1940 S7=50100651
1945 M8=50180832
1950 M9=50200714
1955 M8=49440720
1960 WAIT
1965 USING
1970 INPUT "START? ED";ST
1975 INPUT "ZIEL? ED";ZI
1980 IF ST=0 THEN 1970
1985 IF ZI=0 THEN 1975
1990 PRINT ST
1995 PRINT ZI
2000 AY=INT(ST/10^4)/100
2005 BY=(ST/10^4-AY*100)*100
2010 BR=INT(ZI/10^4)/100
2015 LA=(ZI/10^4-BR*100)*100
2020 A=DEG AY
2025 B=DEG BY
2030 C=DEG BR
2035 D=DEG LA
2040 L=ABS(D-B)
2045 K=SIN A*SIN C+COS A*COS C*COS L
2050 M=ACS K
2055 N=M*111.3
2060 N=INT(N*10+0.5)/10
2065 PRINT "ENTF. (KM)=";N;"KM"
2070 S=(180-C-A+M)/2
2075 IF (S+A-90)<0 THEN 2095
2080 T=SQR((SIN(S-90+A)*SIN(S-M))/((SIN(90-A)*SIN M))
2085 Q=ASN T*2
2090 GOTO 2100
2095 Q=0
2100 IF D>=B THEN 2110
2105 Q=360-Q
2110 Q=INT(Q+0.5)
2115 PRINT "KURS=";Q;"GRAD"
2120 INPUT "NEUES ZIEL?(J/N)";H$
2125 IF H$="J" THEN 1960
2130 IF H$="N" THEN 2140
2135 "B"
2140 PRINT "*** VORHALT-FLUGZEIT ***"
2145 INPUT "V-FLUGZEUG(KT/KMH) 1/2";I$
2150 IF I$="1" THEN 2160
2155 IF I$="2" THEN 2180

```

```

2160 INPUT "V-FLUGZEUG(KT)";R
2165 PRINT "V-FLUGZEUG(KT)";R
2170 R=R*1.852
2175 GOTO 2190
2180 INPUT "V-FLUGZ.(KM/H)=";R
2185 PRINT "V-FLUGZEUG(KM/H)";R
2190 INPUT "KURS=";Q
2195 PRINT "KURS:";Q
2200 INPUT "V-WIND(KT)";T
2205 PRINT "V-WIND(KT)";T
2210 T=T*1.852
2215 INPUT "WINDRICHT.=";U
2220 PRINT "WINDRICHTUNG:";U
2225 A=180+Q-U
2230 B=SIN A
2235 C=T*B/R
2240 D=ASN C
2245 PRINT "VORHALT=";USING "####";D;" GRAD"
2250 GOSUB 3300
2255 IF B=0 THEN 2280
2260 E=180-D-A
2265 F=SIN E
2270 G=R*B/F
2275 GOTO 2300
2280 IF Q=U THEN 2295
2285 G=R+T
2290 GOTO 2300
2295 G=R-T
2300 IF I$="2" THEN 2330
2305 USING
2310 IF I$="1" THEN 2315
2315 G=G/1.852
2320 PRINT "V-KORR.=";USING "###, #";G;"KT"
2325 GOTO 2335
2330 PRINT "V-KORR=";USING "####";G;"KM/STD"
2335 USING
2340 U=N/G
2345 V=DMS U
2350 GOSUB 3350
2355 PRINT "STD:MIN=";USING "##.##";V
2360 USING
2365 INPUT "COUNTDOWN J/N";H$
2370 IF H$="J" THEN 2380
2375 IF H$="N" THEN 2120
2380 GOSUB 3195
2385 PRINT "-----"
2390 "C"
2395 PRINT=LPRINT
2400 PRINT "*** WIND ERFLIEGEN ***"
2405 USING
2410 INPUT "V-FLUGZ.(KM/H)=";VF
2415 PRINT "V-FLUGZ.(KM/H)";VF
2420 INPUT "SOLL-KURS=";Q
2425 PRINT "SOLLKURS:";Q
2430 INPUT "KORR.KURS=";KK
2435 PRINT "KORR.KURS";KK
2440 INPUT "STRECKE(KM)=";ST
2445 PRINT "STRECKE(KM)";ST
2450 INPUT "FLUGZEIT(MIN)";FZ

```

```
2455 PRINT "FLUGZEIT(MIN):";FZ
2460 VT=ST/FZ*60
2465 VW=SQR(SQU VF+SQU VT-2*VF*VT*COS(KK-Q))
2470 VW=VW/1.852
2475 PRINT "V-WIND=";VW;"(KT)"
2480 VW=VW*1.852
2485 AL=ASN(VF/VW*SIN(KK-Q))
2490 IF VT>VF THEN 2500
2495 AL=180-AL
2500 W=Q-AL+180
2505 IF W<360 THEN 2515
2510 W=W-360
2515 IF W>=0 THEN 2525
2520 W=360+W
2525 PRINT "WINDRICHT.:";USING "####";W;"GRD"
2530 INPUT "NEUE RECHNUNG?(J/N)";H$
2535 IF H$="J" THEN 2400
2540 IF H$="N" THEN 2550
2545 PRINT "-----"
2550 "D"
2555 PRINT=LPRINT
2560 PRINT " *** VOR-NAV ***"
2565 TAU=50150810
2570 COL=50470736
2575 WYP=51030717
2580 KIR=49510722
2585 ALB=49131113
2590 ARP=51010818
2595 BAM=51200711
2600 BAY=49591138
2605 BMN=53030847
2610 BKD=53021133
2615 CHA=49550902
2620 HDM=49340828
2625 DKP=49091014
2630 DOM=51430735
2635 DUS=51170645
2640 LBE=53390936
2645 ERD=48201157
2650 ERL=49391109
2655 EUR=47441115
2660 FFM=50030838
2665 FUL=50360934
2670 GED=50250915
2675 GMH=51100754
2680 HAM=53411012
2685 HMM=51510743
2690 HLZ=52221048
2695 DHE=54110755
2700 KRH=48590835
2705 KPT=47451021
2710 KTG=49451013
2715 KBO=50520709
2720 DLE=52150953
2725 LUB=53571040
2730 LBU=48550920
2735 MAH=48161119
2740 MTR=50170851
2745 MIC=54181100
2750 MUN=48111149
```

```
2755 NTM=50010632
2760 NIE=52380922
2765 NUB=49311049
2770 OSN=52120817
2775 RID=49480833
2780 ROD=52130917
2785 RDG=49021232
2790 SAA=49130707
2795 STB=48501235
2800 LBU=48550920
2805 TGO=48370916
2810 SUL=48230839
2815 WLD=48351108
2820 WRB=51300907
2825 WBU=48031205
2830 WSR=5310853
2835 WUR=49430957
2840 NRW=50500642
2845 INPUT "VOR 1 ?";V1
2850 PRINT "VOR 1";V1
2855 INPUT "VOR 2 ?";V2
2860 PRINT "VOR 2";V2
2865 INPUT "FROM VOR 1 ?";P1
2870 PRINT "FROM VOR 1 ?";P1
2875 INPUT "FROM VOR 2 ?";P2
2880 PRINT "FROM VOR 2 ?";P2
2885 P2=P2-2.5
2890 INPUT "ZIELEINGABE ED:";ZI
2895 PRINT "ZIEL-EINGABE:";ZI
2900 WAIT
2905 USING
2910 IF V1=0 THEN 2845
2915 IF V2=0 THEN 2855
2920 IF ZI=0 THEN 2890
2925 B1=INT(V1/TEN 4)/100
2930 L1=(V1/TEN 4-B1*100)*100
2935 B2=INT(V2/TEN 4)/100
2940 L2=(V2/TEN 4-B2*100)*100
2945 BR=INT(ZI/TEN 4)/100
2950 LA=(ZI/TEN 4-BR*100)*100
2955 AV=DEG B1
2960 BV=DEG L1
2965 CV=DEG B2
2970 DV=DEG L2
2975 BZ=DEG BR
2980 LZ=DEG LA
2985 GOSUB 3155
2990 IF Q>180 THEN 3005
2995 R=Q+180
3000 GOTO 3010
3005 R=Q-180
3010 AL=ABS(P1-Q)
3015 IF AL<=180 THEN 3025
3020 AL=360-AL
3025 AM=ABS(R-P2)
3030 IF AM<=180 THEN 3040
3035 AM=360-AM
3040 AN=SIN AM*SIN AL*K-COS AM*COS AL
3045 AN=ACS AN
3050 C1=SIN AM*SIN M/SIN AN
```

```
3055 C1=ASN C1
3060 PRINT "ENTF.VOR1=";C1*111.3
3065 BS=COS(90-AV)*COS C1+SIN(90-AV)*SIN C1*COS P1
3070 BS=ACS BS
3075 GA=SIN C1*SIN P1/SIN BS
3080 GA=ASN GA
3085 LS=BV+GA
3090 GOTO 3100
3095 PRINT "LS=";LS
3100 AV=90-BS
3105 BV=LS
3110 CV=BZ
3115 DV=LZ
3120 GOSUB 3155
3125 N=M*111.3
3130 N=INT(N*10+0.5)/10
3135 PRINT "ENTF.ZIEL";N;"KM"
3140 Q=INT(Q+0.5)
3145 PRINT "KURS ZIEL";Q;"GRAD"
3150 GOTO 2140
3155 K=SIN AV*SIN CV+COS AV*COS CV*COS ABS(DV-BV)
3160 M=ACS K
3165 S=(180-CV-AV+M)/2
3170 T=SQR((SIN(S-90+AV)*SIN(S-M))/(SIN(90-AV)*SIN M))
3175 Q=ASN T*2
3180 IF DV>=BV THEN 3190
3185 Q=360-Q
3190 RETURN
3195 S$=STR$(V+1)
3200 ST$=LEFT$(S$,1)
3205 MI$=MID$(S$,3,2)
3210 SE$=MID$(S$,5,2)
3215 ST=VAL ST$-1
3220 MI=(VAL MI$)*-1
3225 SE=(VAL SE$)*-1
3230 WAIT 46
3235 FOR I=MI TO 0
3240 FOR J=SET 00
3245 USING
3250 PRINT ST;I;J;INT(Q+D)
3255 SE=-59
3260 NEXT J
3265 MI=-59
3270 NEXT I
3275 ST=ST-1
3280 GOTO 3235
3285 WAIT
3290 RETURN
3295 M=G/6
3300 NK=Q+D
3305 IF NK>0 THEN 3340
3310 IF NK<0 THEN 3330
3315 NK=NK-360
3320 PRINT "KORR.KURS:";NK;"GRAD"
3325 RETURN
3330 NK=NK+360
3335 PRINT "KORR.KURS:";NK;"GRAD"
3340 PRINT "KORR.KURS:";NK;"GRAD"
3345 RETURN
3350 M=G/6
```



```
3355 M=M/5
3360 PRINT "10-MIN-MARK";USING "##.#";M;"CM"
3365 G=G*1.852
3370 RETURN
3375 BEEP 3
3380 "K"
3385 BEEP 3
3390 PRINT " KILOMETER/H IN KNOTEN "
3395 INPUT A
3400 B=A*0.540
3405 PRINT A;"KM/H = ";B;" KNOTEN"
3410 INPUT "NEUE BERECHNUNG?(J/N)";H$
3415 IF H$="J" THEN 3390
3420 IF H$="N" THEN 3715
3425 "L"
3430 PRINT "KNOTEN IN KILOMETER/H"
3435 INPUT C
3440 D=C*1.852
3445 PRINT C;"KN = ";D;" KM/H"
3450 INPUT "NEUE BERECHNUNG?(J/N)";H$
3455 IF H$="J" THEN 3430
3460 IF H$="N" THEN 3715
```

Anhang

Programme wurden erstellt
auf SHARP-PC:

KAPITEL 4.1

MATHEMATIK

1. Kurvendiskussion	1401/02
2. Polynomdivision I	1401/02
3. Polynomdivision II	1403
4. Polynommultiplikation	1401/02
5. Interpolation nach Lagrange	1401/02/03 bzw. 1211
6. Gleichungen 3. Grades	1403
7. Lineares Gleichungssystem	1403
8. Relativitätstheorie	1403
9. Statistische Berechnungen	1350/60
10. Bruchapproximation	1401/02
11. Gleichungen 2., 3. und 4. Grades	1401/02
12. Matrizenmultiplikation	1401/02
13. Matrizeninversion	1401/02
14. Matrizenorthogonalisierung	1401/02
15. Integral	1401/02
16. Gauss'scher Algorithmus	1401/02
17. Nullstellen Berechnung	1401/02
18. Differentiation	1401/02
19. Cholesky	1401/02
20. Quadratische Gleichung	1401/02
21. Größter gemeinsamer Teiler	1401/02
22. Kleinstes gemeinsames Vielfaches	1401/02
23. Regula Falsi (Nullstellen)	1401/02
24. Vektorberechnungen	1403
25. Rechnen mit komplexen Zahlen und Gleichungssystemen	1401 mit 10 KB bzw. 1402
26. Finanzmathematik	1403
27. Einlagenrechnung	1401/02
28. Rentenrechnung	1401/02
29. Dreiecksberechnung	1401/02
30. Kreisabschnitt	1401/02
31. Bruchrechnen	1401/02
32. Arithmetische Folgen und Reihen	1401/02
33. Geometrische Folgen und Reihen	1401/02
34. Schiefe Würfe	1401/02

KAPITEL 4.2

ELEKTRONIK UND EDV

1. Dezimal-Binär-Umwandlung	1401/02
2. Dezimal-Hexadezimal-Binär-Umwandlung	1350/60 sowie 1403
3. Hamming-Code	1401/02
4. Wheatstone-Brücke	1401/02
5. Einfaches Schaltinterface	1401/02
6. Oszillator NE 555	1401/02
7. Zenerdiode	1401/02
8. Stern-Dreieck	1401/02
9. Kondensatoren-Berechnung	1401/02
10. Belasteter Spannungsteiler	1401/02
11. Serien-Parallel-Umwandlung	1401/02
12. Schwingkreis	1401/02
13. Vitrometer	1401/02 und 1360

Programme wurden erstellt
auf SHARP-PC:

KAPITEL 4.3

MASCHINENBAU

1. Festigkeitsberechnung	1401/02
2. Momentensatz	1401/02
3. Riemengetriebe	1401/02
4. Brinellhärte	1401/02
5. Statik	1401/02
6. Sparrenberechnung	1401/02

KAPITEL 4.4

SPIEL & SPAB

1. Reaktionstest	1401/02
2. Frogger	1401/02
3. Buchstaben erraten	1401/02
4. Ghostbuster	1401/02
5. Siebzehn und Vier	1401/02
6. Mastermind	1401/02
7. Denktest	1350/60 und andere
8. 51 gleich tot	1350/60
9. Barrikade	1350/60
10. Mondlandung	1350/60
11. Hangman	1401/02
12. Senso	1401/02
13. Shipdown	1401/02
14. Airplane	1403

KAPITEL 4.5

SONSTIGE FACHGEBIETE

1. Biorythmus	1401/02
2. Körperoberfläche	1401/02
3. Stundenplan	1401/02
4. Großschrift mit CE 126	1401/02
5. Namenserkennung	1401/02
6. Mini-Menü	1401/02
7. Telefongebührenzähler	1401/02
8. Digitaluhr	1401/02
9. Laufzeitermittlung	1260/61/62
10. Kalender	1260/61/62
11. Canasta-Abrechnung	1401/02
12. Ilford-Verfahren	1401/02
13. Ilford-Cibachrome-Verfahren	1401/02
14. Tentel-UK3-Prozeß	1401/02
15. Einheitenumrechnung	1401/02 und 1360
16. Chemische Formeln	1350/60
17. Gehäuseberechnung von Lautsprechern	1350
18. Berechnung von aktiven Filtern	1350
19. Reisekonto	1401/02
20. Seenavigation	1401/02
21. Berechnung der Molmasse	1600 und andere
22. Adress-Datei	1403
23. Flugvorbereitung	keine Angabe

Folgende Autoren haben uns Ihre Programme für diese Programmsammlung zur Verfügung gestellt und somit zu deren Gelingen beigetragen:

Klaus Ditze
Paul Debus
Thomas Kindl
Thomas Ocker
Horst Ostholt
Alois Ahlrichs
Rolf Daubmann
Hans Schüeli
Dirk Everling
Thomas Leister
Thomas Aichinger
Wilfried Heller
Daniel Lörtscher
Martin Wozny

Heiko Kieser
Klaus Ellerau
Andreas Klappstein
Markus Schmidt
Hans-Peter Göderz
Lübbo Broers
Stefan Hellmann
Manfred Sohmen
Günter Kolles
Jochen Lührs
Daniel Heid
Hendrik Ambrosius
Karl-Heinz Schneider
Kai Thölen
Falk Langhammer

Die Programm-Sammlung enthält Programme aus vielen verschiedenen Bereichen wie z.B. Mathematik, Physik, Chemie, Maschinenbau, Elektronik, EDV, sonstige Fachgebiete und natürlich auch Spiele.

Mathematik:

- Kurvendiskussion
- Lineare Gleichungssysteme
- Statistische Berechnungen
- Matrizenberechnungen
- Finanzmathematik
- Folgen und Reihen

Maschinenbau:

- Festigkeitsberechnung
- Statik
- Getriebe
- Brinell-Härte

Elektronik/EDV:

- Dezimal-Binär-Hex-Umrechnung
- Wheatstone-Brücke
- Hamming-Code
- Schwingkreis

Spiel und Spaß:

- Reaktionstest
- Ghostbuster
- Frogger
- 17+4
- Mastermind
- Mondlandung
- Airplane

Tips und Tricks:

- Einführung in SHARP-BASIC
- Neue Befehle
- Tips zum Tippen



Yellow Computing · Postfach 1136 · D-7107 Bad Friedrichshall