

VEB ROBOTRON MESSELEKTRONIK  
„OTTO SCHÖN“ Dresden

BASIC-  
Anwender-  
programme  
und  
Hinweise  
zum

**Kleincomputer robotron**  
**KC 85/1**  
**und KC 87**

ausgearbeitet von einem Autorenkollektiv des  
VEB ROBOTRON-Meßelektronik "Otto Schön" Dresden  
unter der Leitung von  
Dr. rer. nat. Gert Keller

Wir danken dem  
VEB Applikationszentrum Elektronik Berlin  
der  
Technischen Universität Dresden, Sektion Mathematik  
und der  
Pädagogischen Hochschule Dresden, Sektion Mathematik  
für die freundliche Unterstützung.

## Vorwort

In der Zeit vom 12. - 15. Mai 1986 fand in Berlin das "Zentrale Fest des Lernens" statt. Im Rahmen dieser Veranstaltung wurden in enger Zusammenarbeit zwischen dem VEB Robotron-Meßelektronik "Otto Schön" Dresden und dem Fernsehen der DDR im Großen Saal des Palastes der Republik etwa 100 Kleincomputer "robotron KC 85/1" zur Nutzung für die teilnehmenden Schüler bereitgestellt.

Besonderer Wert wurde dabei auf die Demonstration der vielseitigen Einsatzmöglichkeiten des Computers gelegt. Zu diesem Zweck standen für die Anwendungsgebiete

- Wissen und Können (Ausbildung)
- Freizeitgestaltung (Spiele)
- Polytechnik (Steuerungsprobleme)
- Textverarbeitung

zahlreiche Programme zur Verfügung, die den Schülern ein schnelles und informatives Arbeiten mit dem Computer ermöglichten.

Ergänzend dazu wurden für die Komplexe

- BASIC-Programmierung
- Assembler-Programmierung

kurze Beispielprogramme erarbeitet, die Hinweise auf Besonderheiten dieser beiden Programmiermöglichkeiten vermittelten.

Die positive Resonanz auf alle bereitgestellten Programme und Texte war uns Anlaß, sämtliche für das "Fest des Lernens" aufbereiteten Bedienungshinweise, Programmbeschreibungen und Beispielprogramme thematisch geordnet in dieser Broschüre zusammenzufassen und einem größeren Kreis von Interessenten zugänglich zu machen.

Besonderer Dank gilt an dieser Stelle den Sektionen Mathematik der TU Dresden und der PH Dresden sowie der Spezialschule Martin-Andersen-Nexö Dresden, die zum Gebiet "Wissen und Können" Programme zur Wissensvermittlung und zum Wissenstest sowie zur Lösung mathematischer Probleme bereitstellten.

Ebenso erhielten wir Unterstützung von den Pädagogischen Hochschulen Güstrow und Halle durch die Vorführung von Programmen und rechnergesteuerten Versuchsabläufen auf dem Gebiet der Polytechnik.

Für Interessenten sei vermerkt, daß alle BASIC-Programme, deren Programmname mit "R+..." beginnt, über den VEB Robotron-Vertrieb Berlin als Software zum Kleincomputer "robotron KC 85/1" erhältlich sind.

Dr. G. Keller  
VEB Robotron-Meßelektronik  
"Otto Schön" Dresden

## Inhaltsverzeichnis

|  | Seite |
|--|-------|
| 1. Hinweise zur Bedienung des Kleincomputers<br>robotron KC 85/1 | 3-6   |
| 2. Programme "Wissen und Können"                                 | 7-36  |
| 3. Programme Freizeitgestaltung                                  | 37-55 |
| 4. Hinweise zur BASIC-Programmierung                             | 56-62 |
| 5. Beispielprogramme BASIC                                       | 63-80 |
| 6. Hinweise zur Programmierung des Plotters XY 4131              | 81-82 |
| 7. Programmierung in Assembler                                   | 83-87 |

# 1. Hinweise zur Bedienung des Kleincomputers robotron KC 85/1

## Starten BASIC

Meldet sich nach dem Einschalten oder nach **RESET** (1) das Betriebssystem mit "OS", so wird durch

BASIC **ENTER**

der BASIC-Interpreter gestartet. Die Anfrage

MEMORY SIZE?:

kann im Normalfall mit **ENTER** beantwortet werden.

Durch



zeigt das BASIC seine Eingabebereitschaft an.

**Achtung!** Soll ein vorher im Speicher vorhandenes BASIC-Programm (z.B. nach **RESET**) erhalten bleiben, so kann der BASIC-Interpreter durch

WBASIC **ENTER**

wieder gestartet werden.

## Laden von BASIC-Programmen von Kassette

Handlungsreihenfolge:

| Schritt | Computer   | Kassettengerät  |
|---------|--|---|
| 1       |  | Programmstart entsprechend Zählerstand aufsuchen. Band kurz vor Vorton (langes "Pfeifen") positionieren |
| 2       | NEW <input type="text" value="ENTER"/>                   |   |
| 3       | CLOAD "prognose"<br>mit konkretem Programmnamen eingeben |   |
| 4       |  | Magnetband starten (Wiedergabetaste > )   |
| 5       | <input type="text" value="ENTER"/>                       | Bei Ertönen des Vortones  |
|         |  | Magnetband stoppen!   |

Danach muß der Cursor etwa je Sekunde eine Position nach rechts rücken. Nach abgeschlossenem Einlesen erfolgt die Meldung "FILE FOUND".

Allgemeine Hinweise zur  
Abarbeitung von BASIC-Programmen

Programmstart

- Erfolgt durch Drücken von **RUN**

Programmabarbeitung

- Programmfortsetzung nach Informationsbildern meist durch **ENTER**
- Bei Zahleneingaben ist ein Dezimalpunkt (statt Komma) zu setzen, z.B.  
123.45 **ENTER**
- Nach der Fehlermeldung "?REDO FROM START" kann eine Eingabe wiederholt werden.
- Häufige Standardantworten bei Alternativfragen sind oft eingeklammert. Im Standardfall genügt dann die Eingabe von **ENTER**  
Beispiel:  
NEUES SPIEL: (J)/N  
Für JA muß nur **ENTER** gedrückt werden.
- Ein BASIC-Programm kann an beliebiger Stelle durch **PAUSE** angehalten (unterbrochen) und durch **CONT** wieder fortgesetzt werden.

Programmende / Programmabbruch

- Normales Programmende wird nur durch "OK" angezeigt.
- Nach Abbruch mit **STOP** erscheint "BREAK IN ...".
- Sollte nach **STOP** nicht der gesamte Bildbereich freigegeben sein, so kann dies durch

WINDOW:CLS **ENTER**

erreicht werden.

### Hinweise zum Drucker

Bei der Initialisierung des Druckers (z.B. nach **RESET**) sind folgende Standardwerte empfehlenswert:

Anzahl der Zeilen: 66  
Anzahl der Leerzeilen: 6  
Zahl der Zeichen: 80

#### Zuschaltung Drucker (im BASIC und Assembler)

Nach der gleichzeitigen Betätigung der Tasten

**CONTR** **P**

wird der Drucker zugeschaltet. Die Bildschirmausgaben (außer "PRINT AT ...") werden danach auch am Drucker ausgegeben. Nochmaliges **CONTR** **P** schaltet den Drucker wieder aus.



## 2. Programme "Wissen und Können"

Die Sammlung der Programme auf dem Gebiet "Wissen und Können" umfaßt Programme verschiedener Schwierigkeitsstufen (praktisch ab 1. Klasse) und verschiedener Wissensgebiete.

Ergänzend zu den Programmen des Herstellerbetriebes (R+MOSAİK, ..., R+MORSET) sind die Programme zur Numerischen Mathematik und Statistik der TU Dresden (R+LINGEN, ..., DISKRI) erfaßt, die fast vollständig im Rahmen der Anwendersoftware zum Kleincomputer angeboten werden.

Dagegen wurden die Programme der PH Dresden und der Martin-Andersen-Nexö-Schule Dresden anläßlich des "Festes des Lernens" erarbeitet.

Interessenten wenden sich deshalb bitte direkt an diese Einrichtungen.

## R+MOSAİK

### Rechenmosaik - Übung der Grundaufgaben

#### Inhaltsbeschreibung

"Rechenmosaik" ist ein Übungsprogramm für 1 bis 3 Personen. Es ist besonders für Schüler der Klassen 1 bis 3 geeignet. Mit dem Programm kann spielend das Rechnen in den 4 Grundrechenarten geübt werden. Der Heimcomputer stellt Aufgaben. Bei deren richtiger Lösung erhält der Spieler einen Mosaikstein auf dem Bildschirm. Falsche Lösungen und Zeitüberschreitung werden vom Rechner angezeigt. Hat ein Spieler sein Mosaik vollständig aufgebaut, wird das Spiel unterbrochen und der Sieger angezeigt. Außerdem gibt der Rechner für jeden Spieler die Anzahl der falschen Lösungen während des Spiels bekannt.

#### Hinweise zur Programmabarbeitung

Am Programmanfang sind im Dialog einzugeben:

- Kennziffer für die gewünschte Rechenart
- Kennziffer für den Zahlenbereich
- Anzahl der Spieler (1 bis 3)
- maximale Rechenzeit je Aufgabe.

Bei eigener Wahl der Grenzen des Zahlenbereichs ist zu beachten, daß die untere Grenze nicht größer als Null und die obere Grenze nicht kleiner als Null sein dürfen.

Abhängig von der Spielerzahl stellt der Rechner ein Zielbild vor. Bei mehreren Spielern besteht dieses aus 2 bzw. 3 Teilbildern. Entsprechend den Spielernummern werden die Teilbilder von links nach rechts zugeordnet.

Rechenzeiten je Aufgabe sind im Bereich 3 bis 30 Sekunden sinnvoll.

Nach der Auswertung eines Spiels kann ein weiteres Spiel folgen oder das Programm abgeschlossen werden. Wenn mit der gleichen Spielerzahl fortgesetzt wird, unterbleibt die Vorstellung des Zielbildes.

# KLEINCOMPUTER robotron KC 85/1

## R+BRUCH1

Demonstrationsprogramm zur Einführung in die Bruchrechnung

### Inhaltsbeschreibung

Anhand grafischer Bilder werden Bruchbegriffe und die Bestandteile eines Bruches veranschaulicht.

Mit der Eingabe selbstgewählter Zahlen in einem gegebenen Wertebereich wird der Obende in die Lage versetzt, sich schnell mit den Voraussetzungen der Bruchrechnung vertraut zu machen.

# KLEINCOMPUTER robotron KC 85/1

## R+BRUCH2

### Übungsprogramm für die Bruchrechnung

#### Inhaltsbeschreibung

Das Programm stellt Aufgaben zum Üben der Addition, Subtraktion, Multiplikation, Division, des Kürzens und Erweiterns gebrochener Zahlen sowie der Hauptnennerrechnung und kontrolliert deren Lösung.

#### Hinweise zur Programmabarbeitung

Die Auswahl der Rechenart erfolgt entsprechend dem Menübild. In jedem Programmteil werden anfangs die Gesetze der Rechenart vorgestellt und anschließend wird der Übenende in seinen Handlungen bis zur Lösung der Aufgabe geführt.

Es werden Anzahl der gelösten Aufgaben und die richtigen Ergebnisse gezählt und am Schluß der Übung das erreichte Ziel ausgewertet.

Die Brüche können sowohl vom Programm vorgegeben, als auch vom Übenenden selbst eingegeben werden, wobei jede Aufgabe nicht mehr als zwei Brüche enthält. Zu große Eingabewerte werden vom Programm zurückgewiesen.

Die aktuelle Eingabeposition wird durch einen dünnen Strich (auch unterstrichener Buchstabe bzw. Zahl) angezeigt.

Die Eingaben sind mit **[ENTER]** abzuschließen. Fragen werden mit

**[J]** (Ja) oder **[N]** (Nein) beantwortet. Einfache Ergebnisse kann man direkt eingeben, wenn zuvor die Taste **[E]** gedrückt wurde. Für die Berechnung komplizierterer Ergebnisse gibt es ein Nebenrechnungsfeld auf dem Bildschirm, auswählbar mit Taste **[N]**.

Dabei ist zu beachten:

- Als Divisionszeichen wird der Doppelpunkt verwendet.
- Die Zahleneingaben werden mit **[ENTER]** abgeschlossen.
- Mit der Taste **[M]** kann ein Ergebnis in einen Speicher überführt werden und eine weitere Nebenrechnung erfolgen.
- Beim Drücken der Taste **[P]** verschwindet das Nebenrechnungsfeld und nach Betätigen der Taste **[E]** kann das Ergebnis eingegeben werden.

Bei Addition und Subtraktion enthalten beide Brüche in der Aufgabenstellung den gleichen Nenner "b" zur Lösung von Aufgaben mit gleichnamigen Brüchen. Die Lösung von Aufgaben mit ungleichnamigen Brüchen wird möglich, wenn für "b" unterschiedliche Zahlen eingegeben werden.

Wurde der Rechner angewiesen, die Brüche auszuwählen, kann vom Übenenden festgelegt werden, ob nur gleichnamige oder auch ungleichnamige Brüche zugelassen werden.

Geometrie - Flächenberechnung

Inhaltsbeschreibung

Das Programm dient der Berechnung von Flächen, wobei zwischen Quadrat, Rechteck, Dreieck und Kreis ausgewählt werden kann. Es werden jeweils verschiedene Formeln zur Berechnung angeboten.

Hinweise zur Programmabarbeitung

Vom Anwender kann in einem ersten Menü (Auswahlbild) die Art der gewünschten Fläche ausgewählt werden. Danach werden jeweils mehrere Formeln zur Flächenberechnung angeboten und die dazu erforderlichen Parameter erläutert. Anschließend fragt das Programm die entsprechenden Parameter ab und prüft diese auf Sinnfälligkeit. Nach vollständiger Eingabe wird der berechnete Flächeninhalt angezeigt.

# KLEINCOMPUTER robotron KC 85/1

## R+HIST01

### Wissenstest - Internationale Geschichte

#### Inhaltsbeschreibung

Das Programm R+HIST01 dient zur Vermittlung, Überprüfung und Festigung von Faktenwissen in Geschichte von der Urzeit bis zur Gegenwart.

#### Hinweise zur Programmabarbeitung

Mittels Menü werden dem Anwender fünf Fragenkomplexe zur Auswahl angeboten. Die Auswahl erfolgt durch Eingabe der Komplexnummer und **ENTER**.

Jede Frage besitzt drei Antworten A, B und C, die tabellenartig auf dem Bildschirm erscheinen. Nur eine davon ist richtig. Die Auswahl erfolgt mittels Tastendruck **A**, **B** oder **C**. Eine Falsch-/Richtig-Aussage und einige Sekunden zum nochmaligen Lesen der richtigen Antwort sind Bestandteil jeder Frage.

Jeder Komplex wird mit einer Auswertung der erreichten Punktzahl abgeschlossen.

# KLEINCOMPUTER robotron KC 85/1

## R+HISTO2

### Wissenstest - Deutsche Geschichte

#### Inhaltsbeschreibung

Das Programm R+HISTO2 dient zur Vermittlung, Überprüfung und Festigung von Faktenwissen in der deutschen Geschichte von der Zeitenwende bis zur Gegenwart.

#### Hinweise zur Programmabarbeitung

Mittels Menü werden dem Anwender fünf Fragenkomplexe zur Auswahl angeboten. Die Auswahl erfolgt durch Eingabe der Komplexnummer und **ENTER**.

Jede Frage besitzt drei Antworten A, B und C, die tabellenartig auf dem Bildschirm erscheinen. Nur eine davon ist richtig. Die Auswahl erfolgt mittels Tastendruck **A**, **B** oder **C**. Eine Falsch-/Richtig-Aussage und einige Sekunden zum nochmaligen Lesen der richtigen Antwort sind Bestandteil jeder Frage.

Jeder Komplex wird mit einer Auswertung der erreichten Punktzahl abgeschlossen.

# KLEINCOMPUTER robotron KC 85/1

R+HIST04

Wissenstest - Geschichte

Frühbürgerliche Revolution und Bauernkrieg

## Inhaltsbeschreibung

Das Programm dient zur Vermittlung, Festigung und Überprüfung von Kenntnissen über die Zeit der Reformation und des deutschen Bauernkrieges.

Eine Karte mit den Grenzen der DDR und BRD und den Flüssen Elbe, Rhein, Main und Donau hilft beim Auffinden von Orten bedeutungsvoller Ereignisse.

Zusätzliche Informationen zu diesen Ereignissen werden bei Bedarf vom Programm ausgegeben.

Am Programmende erfolgt eine Einschätzung der Kenntnisse des Anwenders entsprechend der erreichten Punktezahl.

## Hinweise zur Programmabarbeitung

- Die Beantwortung der Fragen mit vorgegebenen Antworten erfolgt mittels der Tasten A, B, C.
- Entscheidungsfragen (J/N) und Fragen ohne vorgegebene Antworten, wie Fragen nach Orten, Namen werden mit **[ENTER]** abgeschlossen.
- Aus Platzgründen mußte bei der Kartendarstellung und den zugehörigen Antworteingaben auf die **>ENTER<**-Aufforderung verzichtet werden, aber auch hier ist jede Antwort mit **[ENTER]** abzuschließen.



R+AFRI1 , R+AFRI2

Wissenstest Geografie - Afrika

## Inhaltsbeschreibung

Das Wissenstestprogramm dient zur Überprüfung des Wissensstandes sowie zur Festigung und Erweiterung von Faktenwissen. Getestet werden, überwiegend anhand von Umrisskarten, Fakten zu Staaten, Städten und geografischen Besonderheiten Afrikas.

## Hinweise zur Programmabarbeitung

Der Anwender kann vor Beginn des Tests festlegen, ob er (zusätzlich zur mitlaufenden Punktwertung) eine Richtig-/Falsch-Aussage sowie die richtige Lösung im Fehlerfall nach jeder seiner Antworten angezeigt haben möchte oder nicht.

Textliche Eingaben erfolgen in deutscher Schreibweise der geografischen Begriffe und werden mit der Taste **ENTER** abgeschlossen. Zum Zeigen von Punkten auf Karten wird der Cursor auf den betreffenden Punkt positioniert. Dazu werden die Tasten **←**, **→**, **↓**, **↑** benutzt; die Eingabe wird ebenfalls mit der Taste **ENTER** abgeschlossen. Der erreichten Gesamtpunktzahl wird abschließend die entsprechende Bewertungsnote zugeordnet.

# KLEINCOMPUTER robotron KC 85/1

## R+VOKALE

### Lernprogramm für Englisch-Vokabeln

#### Inhaltsbeschreibung

Das Programm unterstützt das Einprägen und Abfragen von Englisch-Vokabeln. Die Vokabeln können über Tastatur oder lektionsweise von der Magnetbandkassette eingegeben werden.

#### Hinweise zur Programmabarbeitung

Alle Vokabeln sind stets orthografisch richtig einzugeben. Das Komma darf nicht verwendet werden.

Es ist möglich bis zu 100 Vokabeln (Worte oder Sätze) in den Computer einzugehen.

## Darstellung mathematischer Funktionen mittels Quasi grafik

Inhaltsbeschreibung

Vom Anwender selbst zu definierende Funktionen der Form  $y = f(x)$  werden ausgewertet und die Bilder der Funktionen werden grafisch dargestellt.

Hinweise zur Programmabarbeitung:

- Nach Start des Programmes wird der Anwender zu Bedienhandlungen aufgefordert. Erfolgte Eingaben sind durch Drücken der Taste **ENTER** abzuschließen.
- Bei der Aufforderung FUNKTIONSEINGABE führt das Programm eine EDIT-Anweisung aus. In der Zeile 1000 ist die darzustellende Funktion mit einer "DEF FN"-Anweisung zu definieren. Der Name der Funktion muß Y sein. Als Funktionsargument sollte die Variable X verwendet werden.

Beispiel:

```
1000 DEF FNY(X)=3+X*(2-7*X)
```

Der Anwender hat seine Funktionseingabe durch Drücken der Taste **ENTER** abzuschließen.

Zur Programmfortsetzung ist die Taste **STOP** zu drücken und nach Erscheinen der Aufforderung ">" die Anweisung

```
GOTO 1000
```

einzugeben.

# KLEINCOMPUTER robotron KC 85/1

R+KIN1

Demonstrationsprogramm für die Gesetze der Kinematik

## Inhaltsbeschreibung

Es werden die Zusammenhänge zwischen Weg, Zeit, Geschwindigkeit, Beschleunigung und die Wurfgesetze mittels anschaulicher Darstellungen und anhand von Funktionsabläufen erläutert.

Der Einfluß der einzelnen physikalischen Größen in den Bewegungsgesetzen kann durch Änderung der Werte innerhalb sinnvoll gewählter Bereiche beobachtet werden.

## Hinweise zur Programmabarbeitung

Mit dem Menübild kann ein Programmteil, der die gewünschte Bewegungsart demonstriert, ausgewählt werden.  
In den Programmteilen 1 bis 3 erfolgt nach dem Demonstrationsteil die Berechnung des entsprechenden Bewegungsablaufs mit Funktionsdarstellung zunächst mit vorgegebenen Werten.  
Die Eingabe selbst gewählter Werte ist möglich, wenn danach auf die Anforderung "Noch einmal" mit J (bzw. nur ENTER) geantwortet wird. Bei der Eingabe ist die unterstrichene Position die jeweils aktuelle Eingabestelle.  
Bei den Programmteilen 4 und 5 bleibt der Demonstrationsteil auf die Veranschaulichung der Geschwindigkeit beschränkt.  
In den Programmteilen 7 und 8 werden zum aktuellen Kurvenverlauf Anfangs- und Endpunkte der früheren Kurvenverläufe als Vergleichsmöglichkeit ausgegeben.

Der Programmteil 8 (schräger Wurf) besteht aus zwei Teilen:  
1. Änderung der Wurfweite in Abhängigkeit vom Wurfwinkel  
2. Abhängigkeit der Wurfweite von der Anfangsgeschwindigkeit

Zu Beginn jedes Teiles können Winkel und Anfangsgeschwindigkeit neu eingegeben werden.

Diesen beiden Teilen liegen unterschiedliche Abbildungsmaßstäbe zugrunde, so daß ein Vergleich des Kurvenverlaufes von Teil 1 mit Teil 2 nicht möglich ist.

Für alle Programmteile gilt:

Die funktionelle Darstellung der Bewegungsabläufe ist für die einzugebenden physikalischen Größen nur in einem ausgewählten Wertebereich aussagefähig. Dieser ist selbst zu ermitteln. Er wird außerdem vom Programm begrenzt, da die Eingabe zu großer Werte zurückgewiesen wird.

# KLEINCOMPUTER robotron KC 85/1

R+KIN2

## Übungsprogramm für die Gesetze der Kinematik

### Inhaltsbeschreibung

Es werden Aufgaben in unterschiedlicher Reihenfolge zu den einzelnen Bewegungsarten und den darin vorkommenden physikalischen Größen gestellt.

Die Anzahl der gewünschten Aufgaben kann gewählt werden. Nach jeder Aufgabe wird bei einer falschen Lösung die richtige angezeigt. Eine Beurteilung der gezeigten Leistungen erfolgt am Übungsende.

### Hinweise zur Programmbearbeitung

Jede Aufgabe enthält in einem ersten Bild die Aufgabenstellung. In einem zweiten Bild erscheint die Darstellung gegeben - gesucht. Die gegebenen Werte werden vom Programm mittels Zufalls-generator ermittelt, so daß sich eine sehr große Zahl von Berechnungsbeispielen ergibt. Als Hilfe bei der Lösung der Aufgaben stehen folgende Funktionen zur Verfügung:

- Taste A - Es kann bei Bedarf wiederholt die Aufgabenstellung angezeigt werden.
- Taste E - Bei einfachen Berechnungen ist sofort die Ergebniseingabe möglich.
- Taste N - Es wird ein Nebenrechnungsfeld auf dem Bildschirm eingeblendet, um kompliziertere Berechnungen nicht erst mit Bleistift und Papier durchzuführen.
- Taste P - Das Nebenrechnungsfeld verschwindet wieder.

Bei der Benutzung des Nebenrechnungsfeldes ist zu beachten:

- Als Divisionszeichen wird der Doppelpunkt verwendet.
- Die Zahleneingaben werden mit **ENTER** abgeschlossen.
- Mit der Taste M kann ein Ergebnis in einen Speicher überführt werden und eine zweite Nebenrechnung erfolgen.

# KLEINCOMPUTER robotron KC 85/1

## R+MORSET

Übungsprogramm für das Hören von Morsetexten

### Inhaltsbeschreibung:

Das Programm eignet sich zum Erlernen und Festigen des Hörens von Morsezeichen im Rahmen der persönlichen Qualifizierung zum Funkamateurlizenznehmer und zur Ausbildung in Arbeitsgemeinschaften "Junge Funker" bzw. im Nachrichtensport der GST.

Das Programm erzeugt nach Eingabe verschiedener und für Morsetexte typischer Parameter über einen Zufallsgenerator Morsetexte in Ser-Gruppen, die auf dem Fernsehbildschirm sichtbar, über die Anschlußbuchse für das Kassettentonbandgerät hörbar gemacht und auch auf Band aufgezeichnet werden können.

### Hinweise zur Programmabarbeitung

Bei der Parametereingabe ist es möglich, mit der Textart 3 spezielle Buchstaben oder Ziffern (1 oder mehrere) anzugeben, die geübt werden sollen. Die Pause zwischen den einzelnen Gruppen kann mit dem Parameter "Pausenverhältnis" variiert werden. (Ø kurze Pause, 6 lange Pause). Durch Drücken einer Taste (außer STOP und RESET) kann die Tonausgabe vorzeitig abgebrochen werden.

Nach jeder Hörübung wird eine Auswahl angeboten. Dabei bedeutet

- 1 - Sprung zum Programmstart
- 2 - Tonausgabe des Textes mit gleichen Parametern
- 3 - Anzeige der alten Parameter, Eingabe neuer Parameter  
Dabei liefert nur Drücken der ENTER-Taste Übernahme des alten Wertes.

!Zur Programmabarbeitung wird eine RAM-Erweiterung benötigt.!

1. Inhalt

Mit diesem Programm können numerische Lösungen der linearen Gleichungssysteme  $Ax = b$  bzw.  $A^{-1}x = b$ , die inverse Matrix zu  $A$  sowie die Determinante von  $A$  berechnet werden. Dabei wird zunächst die quadratische Matrix  $A$  nach einer Variante des Gaußschen Algorithmus mit Spaltenpivotisierung und vollständiger Zeilenäquilibrierung faktorisiert. Danach können die vier genannten Aufgaben in beliebiger Reihenfolge abgearbeitet werden, wobei die sukzessive Behandlung mehrerer rechter Seiten  $b$  möglich ist. Bei der Lösung linearer Gleichungssysteme wird als Maß für die Güte der numerischen Lösung  $\bar{x}$  die betragsgrößte Komponente des Defektvektors  $A\bar{x} - b$  ausgegeben und auf Wunsch eine iterative Nachkorrektur von  $\bar{x}$  vorgenommen.

2. Hinweise zur Benutzung

- Nach der Eingabe der Dimension ( $N \leq 20$ ) der Matrix  $A$  sind die Elemente von  $A$  einzugeben. Ist die Matrix vollständig eingegeben, wird sie blockweise zur Kontrolle tabellarisch aufgelistet, wobei die jeweils angezeigten Elemente korrigiert werden können. Analog erfolgt die Eingabe von  $b$ .
- Die iterative Nachkorrektur einer numerischen Lösung  $\bar{x}$  eines Gleichungssystems wird vom Programm her abgebrochen, wenn sich die letzten beiden Lösungen komponentenweise um weniger als  $5E-6$  betragsmäßig unterscheiden. Aufgrund der geringen Mantissenlänge (3 byte) der Gleitkommazahlen ist bei Algorithmen der linearen Algebra stets mit Rundungsfehlern zu rechnen, so daß auch eine Nachkorrektur u.U. erfolglos arbeitet.

3. Beispiel

Drei Zahnräder eines Getriebes haben zusammen 80 Zähne. Bei 10 Umdrehungen des ersten Rades drehen sich das zweite 18 und das dritte 45mal. Wieviel Zähne hat jedes Rad?

Modell:  $x_i$  = Anzahl der Zähne des Rades  $i$ ;  $i=1,2,3$

$$\begin{array}{rcl} x_1 + x_2 + x_3 & = & 80 \\ 10x_1 - 18x_2 & = & 0 \\ 10x_1 & - & 45x_3 = 0 \end{array} \Rightarrow A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1.8 & 0 \\ 1 & 0 & -4.5 \end{bmatrix}, b = \begin{bmatrix} 80 \\ 0 \\ 0 \end{bmatrix}.$$

1. Inhalt

Es kann eine numerische Lösung eines linearen Gleichungssystems  $Ax = b$  mit symmetrischer Matrix  $A = A^T$  nach dem Algorithmus von BUNCH/PARLETT berechnet werden, wobei die positive Definitheit von  $A$  hier nicht verlangt wird (d.h.  $A$  kann positive und auch negative Eigenwerte besitzen).

Nach einmaliger  $LDL^T$ -Faktorisierung von  $A$  können sukzessive mehrere rechte Seiten  $b$  bearbeitet werden, wobei jeweils eine iterative Nachkorrektur der Näherungslösungen möglich ist. Zudem können die Determinante von  $A$  und Eigenwertcharakteristik (Anzahl der positiven und negativen Eigenwerte) von  $A$  bestimmt werden.

2. Hinweise zur Benutzung

- Nach der Eingabe der Dimension ( $N \leq 20$ ) der Matrix  $A$  sind die Elemente der oberen Dreiecksmatrix von  $A$  einzugeben. Anschließend wird die obere Dreiecksmatrix tabellarisch aufgelistet, wobei die jeweils angezeigten Matrixelemente auf Wunsch korrigiert werden können. Analog erfolgt die Eingabe der rechten Seite  $b$ .
- Zu jeder numerischen Lösung  $\bar{x}$  wird die betragsgrößte Komponente des Defektvektors  $A\bar{x} - b$  ausgegeben. Der Anwender kann nun entscheiden, ob eine iterative Nachkorrektur erforderlich ist. Das Programm lehnt die Nachkorrektur ab, falls sich die letzten beiden Lösungen betragsmäßig um weniger als  $5E-6$  in den Komponenten unterscheiden. Aufgrund der geringen Mantissenlänge (3 byte) der Gleitkommazahlen ist bei Verfahren der linearen Algebra stets mit Rundungsfehlern zu rechnen, so daß auch eine Nachkorrektur u.U. erfolglos bleibt.

3. Beispiel

Die Städte Dresden, Leipzig und Berlin bilden die Eckpunkte eines Dreieckes. Von Dresden über Berlin nach Leipzig beträgt die Entfernung 406 km, von Leipzig über Dresden nach Berlin 303 km und von Berlin über Leipzig nach Dresden 323 km.

Man berechne die Entfernungen Dresden-Leipzig ( $x_1$ ),  
 Dresden-Berlin ( $x_2$ ),  
 Leipzig-Berlin ( $x_3$ ).

Modell:

$$\begin{array}{rcl}
 & x_2 + x_3 & = 406 \\
 x_1 + x_2 & & = 303 \\
 x_1 & + x_3 & = 323
 \end{array}
 \Rightarrow A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 406 \\ 303 \\ 323 \end{bmatrix}$$



LINEARE ÜBERBESTIMMTE GLEICHUNGSSYSTEME1. Inhalt

Es kann eine Quadratmittellösung eines überbestimmten Gleichungssystems  $Ax = b$  ( $A$  hat  $N$  Spalten und  $M \geq N$  Zeilen) mittels eines implizit realisierten GIVENS-Verfahrens berechnet werden. Neben der elementweisen Eingabe von  $A$  und  $b$  können  $A$  und  $b$  auch aus einer linearen Regressionsaufgabe durch das Programm generiert werden. Dazu sind  $N$  Ansatzfunktionen  $q_k(t)$ ,  $k=1,2,\dots,N$ , zu definieren und  $M (\geq N)$  Punktepaare  $(t_j, y_j)$ ,  $j=1,2,\dots,M$ , einzugeben, die dem funktionalen Zusammenhang

$$y = x_1 \cdot q_1(t) + x_2 \cdot q_2(t) + \dots + x_N \cdot q_N(t)$$

genügen sollen. Die Parameter  $x_1, x_2, \dots, x_N$  werden dann im Sinne der Gaußschen Fehlerquadratmethode bestimmt.

2. Hinweise zur Benutzung

- Bei der direkten Eingabe der  $(M,N)$ -Matrix  $A$  und des Vektors  $b$  ist nach der elementweisen Eingabe eine Korrektur der nochmals aufgelisteten Elemente möglich.
- Im Fall der linearen Regression sind die Ansatzfunktionen  $q_k(t)$ , die von den zu schätzenden Parametern  $x_1, x_2, \dots, x_N$  nicht abhängen dürfen, im automatisch eingestellten EDIT-Modus als  $Q(1) = q_1(t), \dots, Q(N) = q_N(t)$  in die Programmzeilen 1000 bis 1100 einzutragen. Nach dem Rücksprung in das Programm durch die Anweisung GOTO 100 werden Zahlen für  $N$  und  $M$  sowie die Punktepaare  $(t_j, y_j)$ ,  $j=1,2,\dots,M$  abgefordert. Eine Korrektur der Punktepaare ist abschließend bei Bedarf möglich.

3. Beispiel

Bei einem Wurf senkrecht nach oben besteht zwischen dem Weg  $y$  und der Zeit  $t$  die Beziehung  $y = c \cdot t - 0.5 \cdot g \cdot t^2$ , wobei  $c$  die Anfangsgeschwindigkeit und  $g$  die Fallbeschleunigung ist. Man beurteile, ob die Meßreihe

|      |   |      |      |      |      |      |     |
|------|---|------|------|------|------|------|-----|
| T(I) | 0 | 1    | 2    | 3    | 4    | 5    | 6   |
| Y(I) | 0 | 25.1 | 40.4 | 45.9 | 41.6 | 27.5 | 3.6 |

einem Wurf senkrecht nach oben entspricht. Wie groß ist die Anfangsgeschwindigkeit?

Modell:  $X(1) \hat{=} c$ ,  $X(2) \hat{=} 0.5 \cdot g \approx 4.9$ ,  $N = 2$ ,  $M = 7$

Definition der Ansatzfunktionen  $q_1(t) = t$ ,  $q_2(t) = -t^2$ ;  
 $\Rightarrow$  1000  $Q(1) = T$  :  $Q(2) = -T \cdot T$ .

1. Inhalt

Das Programm berechnet nacheinander alle Nullstellen eines Polynoms n-ten Grades mit reellen bzw. komplexen Koeffizienten nach einem Verfahren von NICKEL (NEWTON-Verfahren für komplexe Polynome). Anfangsnäherungen für die einzelnen Nullstellen brauchen vom Anwender nicht bereitgestellt zu werden. Eine graphische Veranschaulichung der Lage der Nullstellen in der komplexen Zahlenebene ist nach der Rechnung möglich.

2. Hinweise zur Benutzung

- Nach der Eingabe des Polynomgrades ( $N \leq 50$ ) sind die einzelnen Koeffizienten einzugeben. Anschließend besteht eine Korrekturmöglichkeit der angezeigten Koeffizienten.
- Sobald eine Nullstelle berechnet ist, wird sie zahlenmäßig ausgegeben. Wenn alle Nullstellen bekannt sind, kann eine Darstellung der Lage dieser Nullstellen in der komplexen Zahlenebene erfolgen. Danach ist eine erneute wertmäßige Ausgabe der Nullstellen möglich.
- Besitzt das Polynom betragsmäßig sehr große Koeffizienten, kann während der Rechnung Zahlenbereichsüberlauf (OV-Error) auftreten. In diesem Fall sollte das Polynom normiert werden.
- Aufgrund der geringen Mantissenlänge (3 byte) der Gleitkommazahlen treten bei mehrfachen Nullstellen merkliche Rundungsfehler auf.

3. Beispiel

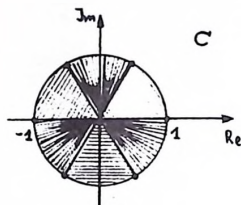
Eine durch einen Einheitskreis begrenzte Scheibe soll in  $N (= 6)$  gleiche Teile eingeteilt werden. In der komplexen Ebene  $C$  wird der Einheitskreis durch die Gleichung  $x^N = 1$ ,  $x \in C$ , beschrieben. Die gleiche Teilung der Einheitskreislinie in  $N$  Teile wird durch die  $N$  Nullstellen des Polynoms  $p_N(x) = x^N - 1$  realisiert.

Modell:  $N = 6$

reelle Koeffizienten:

$$A(N) = 1, A(0) = -1,$$

$$A(N-1) = \beta, \dots, A(1) = \beta.$$



1. Inhalt

Das Programm berechnet für eine auf dem Intervall  $[a,b]$  definierte Funktion  $f$  eine Nullstelle. Nach der Festlegung von  $f$  als Nutzerfunktion und der Eingabe von  $a$  und  $b$  kann die Funktion zunächst skizziert werden (Print-Plot). Zur Nullstellenbestimmung stehen fünf ableitungsfreie Verfahren zur Auswahl:

- (1) eine Nullstellen einschließende, konvergenzbeschleunigte Regula-falsi-Methode (Pegasus-Algorithmus)
- (2) eine modifizierte Regula-falsi für mehrfache Nullstellen
- (3) ein Minimumsuchalgorithmus für den Betrag von  $f$
- (4) ein Intervallhalbierungsverfahren
- (5) ein Algorithmus zur inversen quadratischen Interpolation

Das Programm nimmt auf Wunsch automatisch eine Verfahrensauswahl vor.

2. Hinweise zur Benutzung

- Bei der Aufforderung FUNKTIONSDEFINITION stellt das Programm den EDIT-Modus ein. In der Zeile 100 ist die Funktion  $f$  mit einer DEF FN-Anweisung festzulegen. Der Name der Funktion ist  $F$ .
- Nach Beendigung der Eingabe von  $f$  muß der EDIT-Modus mittels STOP beendet und die Fortsetzung des Programms mittels GOTO 100 realisiert werden.
- Wenn die Funktion skizziert wird, ermittelt das Programm ein Einschließungsintervall für die der unteren Intervallgrenze  $a$  am nächsten gelegenen Nullstelle.
- Die Verfahren (1),(2),(4) und (5) sind an die Voraussetzung  $f(a) \cdot f(b) < 0$  gebunden.

3. Beispiel

Man bestimme die Schnittpunkte der Graphen der Funktion  $y = \sin x$  und  $y = e^{-x}$  im Intervall  $[0,4]$ .

Modell:  $f(x) = \sin(x) - e^{-x}$ ,  $x \in [0,4]$

$\Rightarrow$  100 DEF FNF(X) = SIN(X) - EXP(-X).

NICHTLINEARE (ÜBERBESTIMMTE) GLEICHUNGSSYSTEME

1. Inhalt

Es kann eine Quadratmittellösung eines nichtlinearen Systems aus  $m$  Gleichungen  $F_1(x) = 0, \dots, F_m(x) = 0$  in  $n$  ( $\leq m$ ) Unbekannten  $x = (x_1, \dots, x_n)$  nach einem gedämpften GAUSS-NEWTON-Verfahren (Abstiegssprinzip) berechnet werden. Dabei lassen sich entweder die  $m$  Funktionen  $F_1, \dots, F_m$  direkt in das Programm eintragen oder eine von den  $n$  zu bestimmenden Parametern  $x_1, \dots, x_n$  abhängende Ansatzfunktion  $Q(t, x_1, \dots, x_n)$  definieren und  $m$  Punktepaare  $(t_j, y_j), j=1, \dots, m$ , eingeben, die dem Zusammenhang  $y_j = Q(t_j, x_1, \dots, x_n)$  genügen sollen. Im Fall  $m = n$  wird das gedämpfte NEWTON-Verfahren realisiert.

2. Hinweise zur Benutzung

- Nach der Festlegung der Aufgabenklasse (Nichtlineare Regression oder Gleichungssystem) stellt das Programm den EDIT-Modus ein. In den angezeigten Zeilen sind dann entweder die Ansatzfunktion  $Q = Q(t, x_1, \dots, x_n)$  oder die Funktionen  $F(1) = F_1(x_1, \dots, x_n), \dots, F(m) = F_m(x_1, \dots, x_n)$  mit  $X(1) = x_1, \dots, X(n) = x_n$  zu definieren. Dazu können vom Anwender auch einbuchstabige Hilfsgrößen verwendet werden.
- Nach dem Rücksprung in das Programm durch Betätigen der Tasten STOP und RUN werden die Zahlen  $m$  und  $n$  sowie im Fall der Regression die Punktepaare  $(t_j, y_j)$  abgefordert. Außerdem ist eine Startnäherung für die zu ermittelnden  $X(1), \dots, X(n)$  einzugeben.
- Während der Rechnung wird zur Beobachtung des Verfahrens ein Protokoll mit charakteristischen Größen (Norm des Defektvektors, Gradientennorm) ausgegeben.
- Bei stark nichtlinearen Problemen bzw. bei ungünstigen Startwerten kann das Verfahren vorzeitig abbrechen. Aufgrund der Rechengenauigkeit des Computers sollten nur schwach nichtlineare Aufgaben bearbeitet werden.

3. Beispiel Vom Element Radon wurde der Zufallsprozeß gemessen.

|              |     |     |     |     |      |
|--------------|-----|-----|-----|-----|------|
| Zeit $t$     | 1.2 | 6.5 | 7.7 | 8.4 | 14.0 |
| Kernzahl $y$ | 82  | 31  | 25  | 22  | 8    |

Man berechne die Anzahl  $X(1) = N_0$  der ursprünglichen Atomkerne und die Zerfallskonstante  $X(2) = \lambda$  aus dem Zerfallsgesetz  $y = N_0 \cdot e^{-\lambda t}$ .

Modell: Ansatzfunktion  $Q = X(1) \cdot \text{EXP}(-X(2) \cdot T)$ .

1. Inhalt

Das Programm kann punktweise eine Lösungskurve eines parameterabhängigen nichtlinearen Systems aus  $n$  Gleichungen  $H_1(u) = 0$ ,  $H_2(u) = 0$  in  $n+1$  Unbekannten  $u = (u_1, \dots, u_n, u_{n+1})$  nach einem Prädiktor-Korrektor-Verfahren (Einbettungstechnik) berechnen. Im EDIT-Modus sind dazu die Funktionen  $H_1, \dots, H_n$  direkt in das Programm einzutragen. Als Korrektor wird ein diskretisiertes NEWTON-Verfahren verwendet. Nachdem die Lösungskurve im bezüglich  $t = u_{n+1}$  vorgegebenem Intervall berechnet worden ist, lassen sich einzelne Komponenten der Lösungskurve gegenüber  $t$  graphisch darstellen.

2. Hinweise zur Benutzung

- Zunächst werden die Funktionen  $H_i$  in der Form  $H(1) = H_1(u), \dots, H(u) = H_n(u)$  in die vom Programm angezeigten Zeilen eingetragen. Nach dem Rücksprung in das Programm durch Betätigen der Tasten STOP und RUN erfolgt die Eingabe des Verfahrensparameters:
 

|        |  |                      |
|--------|--|----------------------|
| N(= n) | Anzahl der Gleichungen   |                      |
| TA     | untere Intervallgrenze für $t = u_{n+1}$                                       | [Standard: TA = -1]  |
| TE     | obere Intervallgrenze für $t = u_{n+1}$  | [Standard: TE = 1]   |
| TAUØ   | Anfangsschrittweite  | [Stand.: TAUØ = Ø.1] |
| TAUM   | maximale Schrittweite  | [Stand.: TAUM = 1]   |
| EPS    | Abbruchschranke für die Korrekteriteration                                     | [Stand.: EPS = 1E-4] |
| TS     | ausgezeichneter Lösungspunkt mit $u_{n+1} = t = TS$                            |                      |
| B(= b) | Index $b \in \{1, 2, \dots, n\}$ ; $u_b$ wird während der Rechnung ausgegeben. |                      |
- Anschließend ist ein Startvektor  $u = (u_1, \dots, u_{n+1})$  mit der Eigenschaft  $H_1(u) = H_2(u) = \dots = H_n(u) = 0$  einzugeben.
- Während der Rechnung werden die aktuellen Werte von  $t, u_b$ , die Norm des Defektvektors ( $H_1(u), \dots, H_n(u)$ ) ausgegeben. Es ist möglich, die Größen TAUM, EPS und B während der Rechnung zu modifizieren.

3. Beispiel

Das nichtlineare Gleichungssystem  $F_1(x) = 0, F_2(x) = 0, x = (x_1, x_2)$  soll mittels Einbettung  $H_1(x, t) = F_1(x) - (1-t)F_1(x^0), x^0$  bekannt, gelöst werden.

$$F_1(x) = x_1 + 5x_2 \cdot x_2 - 2x_2 - x_2 \cdot x_2 - 13, x_1^0 = 15, F_1(x^0) = 34$$

$$F_2(x) = x_1 + x_2 \cdot x_2 - 14x_2 + x_2 \cdot x_2 - 29, x_2^0 = -2, F_2(x^0) = 10$$

Modell:  $H(1) = U(1) - 2 \cdot U(2) + 5 \cdot U(2) \cdot U(2) - U(2) \wedge 3 + 34 \cdot U(3) - 47$

$$H(2) = U(1) - 14 \cdot U(2) + U(2) \cdot U(2) + U(2) \wedge 3 + 1Ø \cdot U(3) - 39$$

$$N = 2, TS = 1, B = 2 \text{ (Übrige Parameter gemäß Standard).}$$

1. Inhalt

Die Varianzanalyse ist ein statistisches Verfahren zur quantitativen Untersuchung von Einflüssen (Effekten) eines oder mehrerer Faktoren auf Versuchsergebnisse. Das vorliegende Programm ist für Modelle mit "festen Effekten" ausgelegt. Dabei kann für eine Einflußgröße die EINFACHE KLASSIFIKATION mit gleicher oder ungleicher Anzahl von Wiederholungen und im Falle zweier Einflußgrößen die KREUZKLASSIFIKATION verwendet werden. Bei der Kreuzklassifikation sind folgende Varianten möglich:

- a) ohne Wechselwirkung (d.h. ohne Wiederholungen)
- b) mit Wechselwirkung und gleicher Anzahl von Wiederholungen
- c) mit Wechselwirkung und ungleicher Anzahl von Wiederholungen

2. Hinweise zur Benutzung

- Nach Auswahl der Klassifikation werden die Anzahl der Stufen und Wiederholungen sowie anschließend (wahlweise über Tastatur oder von Kassette) die Meßwerte eingegeben.
- Fehlermeldungen erscheinen bei ungeeigneten Datenstrukturen.

3. Beispiel

Zur Untersuchung der Ertragsfähigkeit einer Weizensorte wurden drei Versuchsflächen A, B und C mit unterschiedlichen Düngermengen versehen. Stichproben - entnommen von allen drei Flächen - ergaben folgende Ertragswerte.

|                | Wiederholungen |     |     |     |     |          |
|----------------|----------------|-----|-----|-----|-----|----------|
|                | 1              | 2   | 3   | 4   | 5   |          |
| Fläche A       | 108            | 110 | 106 | 113 |     | M = 3    |
| Stufe Fläche B | 98             | 101 | 104 | 110 | 104 | S(1) = 4 |
| Fläche C       | 112            | 109 | 123 | 118 |     | S(2) = 5 |
|                |                |     |     |     |     | S(3) = 4 |

(Behandlung: EINFACHE KLASSIFIKATION mit ungleicher Anzahl von Wiederholungen). Bei Vorgabe von  $\alpha = 0,05$  läßt sich wegen  $T = 7,31 > F_{1-\alpha, 2, 10} = 4,10$  feststellen, daß die Düngermenge einen (statistisch gesicherten) Einfluß auf den Ertrag hat.)

1. Inhalt

Die Clusteranalyse ist ein (i.a. heuristisches) Verfahren zur Aufteilung von Objekten in Gruppen (Cluster). Das Programm enthält drei verschiedene Algorithmen (agglomeratives, divisives, Leader-Verfahren). Bei agglomerativen Verfahren wird von  $M$  zu klassifizierenden Objekten ausgegangen, die  $M$  einelementigen Clustern entsprechen, und schrittweise jeweils die Cluster mit dem minimalen Abstand vereinigt, bis sämtliche Elemente in einem Cluster enthalten sind. Bei divisiven Verfahren wird der entgegengesetzte Weg beschritten, d.h., es werden - ausgehend von einem alle Elemente enthaltenden Cluster - durch fortlaufende Aufspaltung Cluster erzeugt, bis schließlich jedes Cluster nur ein Element enthält. Beim Leader-Verfahren werden ausgehend von  $N$  vorzuziehenden Führungselementen (Leader) die anderen Objekte derart gruppiert, daß die Abstandskadratsumme innerhalb der Cluster minimal wird.

2. Hinweise zur Benutzung

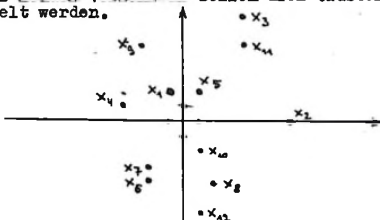
- Das Programm kann maximal 45 Objekte mit 12 Variablen verarbeiten.
- Zur Erklärung der Abstandsfunktionen siehe Programmdokumentation.

3. Beispiel

Gegeben seien 12 Objekte mit 2 Variablen, die sich in folgender Weise in der Ebene darstellen lassen:

- $x_1 = (-1, 2), \quad x_7 = (-2, -3)$   
 $x_2 = (7, 0), \quad x_8 = (2, -4)$   
 $x_3 = (4, 7), \quad x_9 = (-3, 5)$   
 $x_4 = (-4, 1), \quad x_{10} = (1, -2)$   
 $x_5 = (1, 2), \quad x_{11} = (4, 5)$   
 $x_6 = (-2, -4), \quad x_{12} = (1, -6)$

Mittels des Leader-Verfahrens sollen drei Cluster (Leader: 3, 6, 11) ermittelt werden.



1. Inhalt

Es wird die Hypothese geprüft, daß die (unbekannte) Verteilungsfunktion  $F$  einer Grundgesamtheit, der eine Stichprobe vom Umfang  $N$  entnommen wurde, gleich einer vorgegebenen hypothetischen Verteilungsfunktion  $F_0$  ist. Die Testgröße  $TG$ , die aus der maximalen Abweichung der hypothetischen von der empirischen Verteilungsfunktion gebildet wird, ist asymptotisch kolmogorov-verteilt.

2. Hinweise zur Benutzung

- In einem Menübild werden verschiedene hypothetische Verteilungsfunktionen (Normalverteilung, gleichmäßige Verteilung, Weibullverteilung, Exponentialverteilung) sowie die Möglichkeit der nutzer eigenen Definition einer beliebigen stetigen Verteilungsfunktion  $F_0(t)$  angeboten.
- Fehlermeldungen erscheinen bei ungeeigneten Datenstrukturen; Fehler, die aus einer vom Nutzer falsch definierten Verteilungsfunktion oder der Vorgabe "sehr ungeeigneter" Parameter für die hypothetische Verteilungsfunktion resultieren, werden i.a. nicht abgefangen.
- Sämtliche Hilfsgrößen in der vom Nutzer zu definierenden Verteilungsfunktion müssen mit dem Buchstaben T beginnen.

3. Beispiel

Bei der Messung des Gewichtes von 25 Äpfelbeuteln wurden folgende Daten erhalten:

1010, 970, 985, 970, 990, 985, 1030, 940, 930, 960,  
975, 995, 1005, 1010, 960, 945, 925, 960, 975, 980,  
990, 1005, 995, 930, 955

Es soll die Hypothese geprüft werden, daß das Gewicht der Äpfelbeutel normalverteilt mit Mittelwert  $EX = 1000$  und Streuung  $DX^2 = 625$  ist. Dabei wird  $\alpha = 0.05$  gewählt. Als Testgröße erhält man  $TG = 0,379$  und damit die Ablehnung der Hypothese.



1. Inhalt

Auf der Grundlage des durch BASIC bereitgestellten Zufallszahlengenerators RND für gleichmäßig auf dem Intervall  $(0, 1)$  verteilte Zufallszahlen werden für folgende Verteilungen Zufallszahlen erzeugt:

- a) beliebige gleichmäßige Verteilung
- b) Normalverteilung
- c) Exponentialverteilung
- d) Weibullverteilung
- e) Binomialverteilung

2. Hinweise zur Benutzung

- Bei der Eingabe der Verteilungsparameter sind die entsprechenden Voraussetzungen (z.B. Positivität) zu beachten. Der Parameter N der Binomialverteilung ist auf  $N = 500$  beschränkt.
- Bei ungeeigneten Parameterkonstellationen können OV-Errors auftreten. In einem solchen Fall ist die Rechnung zu wiederholen.
- Es können maximal 500 Zufallszahlen erzeugt werden.

3. Beispiel

Es sollen 25 normalverteilte Zufallszahlen mit den Parametern Mittelwert  $M = 10$ , Streuung  $S = 4$  erzeugt werden. Eine Reduzierung der Dezimalstellen soll nicht erfolgen.

1. Inhalt

Die Korrelationsanalyse untersucht stochastische Zusammenhänge zwischen  $K$ , ( $K \geq 2$ ) Zufallsgrößen anhand einer Stichprobe. Für jedes Paar von Zufallsgrößen werden die einfachen Produkt-Moment-Korrelationskoeffizienten berechnet. Wahlweise ist zusätzlich die Berechnung der Bestimmtheitsmaße sowie ausgewählter (bzw. sämtlicher) partieller und multipler Korrelationskoeffizienten (falls mehr als 2 Zufallsgrößen untersucht werden) möglich.

2. Hinweise zur Benutzung

- Die Berechnung der partiellen und multiplen Korrelationskoeffizienten ist ggf. recht zeitaufwendig.
- Bei linearen Abhängigkeiten in der Kovarianzmatrix können gewisse partielle und multiple Korrelationskoeffizienten nicht berechnet werden.
- Bei der Dateneingabe von Kassette sind die Bildschirmaufforderungen genau zu befolgen.

3. Beispiel

Bei einer chemischen Reaktion werden an drei verschiedenen Messstellen (entspricht 3 Merkmalen) jeweils 4 Messungen durchgeführt. Es ergaben sich die folgenden Werte:

| Messung | Merkmal |   |   |
|---------|---------|---|---|
|         | 1       | 2 | 3 |
| 1       | 1       | 6 | 4 |
| 2       | 3       | 2 | 4 |
| 3       | 2       | 1 | 3 |
| 4       | 4       | 9 | 5 |

Zur Berechnung des Grades der linearen Abhängigkeit der drei Merkmale untereinander sollen die einfachen und sämtliche partiellen und multiplen Korrelationskoeffizienten bestimmt werden.

1. Inhalt

Die Diskriminanzanalyse ist ein Verfahren zur Bestimmung optimaler Trennfunktionen zwischen Kollektiven zum Zwecke der Zuordnung von Objekten unbekannter Herkunft zu einem der Kollektive. Neben der Berechnung der Trennfunktionen - auf der Basis von Objektmengen für jedes der Kollektive - wird der Abstand zwischen den Kollektiven (im Sinne von Mahalanobis) ermittelt und die Hypothesen auf (paarweise) Übereinstimmung der Kollektive getestet. Auf Grund dieser Kenndaten wird für jedes eingegebene Objekt unbekannter Herkunft versucht, die Zuordnung zu einem der Kollektive zu treffen.

2. Hinweise zur Benutzung

- Das Programm ist für maximal 20 Kollektive oder 50 Merkmale oder 50 Objekte pro Kollektiv ausgelegt;
- Fehlermeldungen erscheinen bei zu geringem Stichprobenumfang bzw. bei singulärer Kovarianzmatrix

3. Beispiel

Die Produktion eines Teiles wurde in drei Güteklassen ( $Q = 3$ ) A ( $\hat{=}$  gut), B ( $\hat{=}$  mittel), C ( $\hat{=}$  schlecht) eingeteilt. Es stehen folgende Stichprobenobjekte (mit jeweils 2 meßbaren Merkmalen ( $N = 2$ )) zur Verfügung, die über Tastatur einzugeben sind.

Klasse A: (6 Objekte,  $P(1) = 6$ )

$$x_1 = (4, 3), x_2 = (7, 9), x_3 = (-8, 3), x_4 = (-3, 4), \\ x_5 = (5, 5), x_6 = (-1, 2)$$

Klasse B: (5 Objekte,  $P(2) = 5$ )

$$y_1 = (6, -2), y_2 = (9, 1), y_3 = (8, 0), y_4 = (12, -2) \\ y_5 = (4, -3)$$

Klasse C: (4 Objekte,  $P(3) = 4$ )

$$z_1 = (-8, 19), z_2 = (-9, 10), z_3 = (-10, 22), z_4 = (-7, 15)$$

Das Objekt  $t = (-3, 5)$  unbekannter Herkunft soll einem der Kollektive A, B oder C zugeordnet werden.

# KLEINCOMPUTER robotron KC 85/1

## U-ADDEZ

- Inhalt:** Mit Hilfe dieses Programms können einzelne Schüler (oder andere Nutzer) ihre Fertigkeit im Addieren von Dezimalbrüchen trainieren. Nach nur kurzer Einleitung und einigen speziellen Nutzerhinweisen werden wahlweise 5 oder 10 verschiedene Aufgaben mit zunehmendem Schwierigkeitsgrad gestellt. Die Lösungen werden kommentiert und mit Punkten bewertet, bei falschen Ergebnissen wird auf mögliche Fehlerursachen hingewiesen. Zum Schluß erfolgt eine Benotung.
- Hinweise:** Die Generierung der Aufgaben erfolgt im Rahmen des entsprechenden Schwierigkeitsgrades zufällig. Vom Nutzer wird nur die Eingabe der Lösung verlangt. Fehleingaben werden mit entsprechendem Kommentar abgelehnt. Falls gewünscht, ist ein vorzeitiger Abbruch des Programms möglich.
- Programm und Beschreibung:** PH Dresden

## D-VARI

### 1. Inhaltsbeschreibung:

Das Programm "D-VARI" demonstriert in anschaulicher Art und Weise die Anwendung und Verarbeitung von Variablen im Kleincomputer als Paar von festem Namen und veränderlichem Wert. Behandelt werden numerische, Zeichenketten- und Feldvariablen.

Es handelt sich um ein reines Demonstrationsprogramm, weil dies zum Selbststudium des Programmierens am Computer für den Unterricht in der allgemeinbildenden Oberschule und der Berufsausbildung sowie für die Lehre an den Hoch- und Fachschulen geeignet ist. Außerdem leistet es einen Beitrag zum Verständnis von Grundlagen der Informatik - die dynamische Ergibtanweisung

$A:=A+1$

anstelle einer Gleichung im mathematischen und statischen Sinne.

### 2. Hinweise zur Programmabarbeitung:

Nach Start des Programms erfolgt eine kontinuierliche Abarbeitung, ohne daß der Anwender aktiven Einfluß nimmt.

|               |       |                                    |
|---------------|-------|------------------------------------|
| Variablenname | ————→ | Quader                             |
| Wertzuweisung | ————→ | Wanderung des Wertes in den Quader |

Programm und Beschreibung: PH Dresden

## S-PHYSIK

**Inhalt:** Es wird die Verteilung von  $N$  Teilchen auf die beiden Hälften eines Volumens simuliert. Diese Simulation wird 2mal wiederholt und die Häufigkeit des Auftretens der möglichen verschiedenen Verteilungen in einem Balkendiagramm dargestellt. Dieses Programm ist zweckmäßig zur Veranschaulichung des statistischen Wesens der Thermodynamik im Schulunterricht in Klasse 11 einzusetzen.

**Hinweise:** Die Zahl  $N$  der Teilchen sollte zwischen 7 und 30 liegen, wobei kleinere  $N$  vorzuziehen sind. Bei der Auswahl von  $Z$  ist der Zielfaktor zu berücksichtigen. Bei der graphischen Darstellung der Verteilung läßt sich die Verweilzeit problemlos ändern. Dazu muß in Zeile 1705 die Zahl hinter der Anweisung PAUSE variiert werden, eine Einheit entspricht etwa einer Zehntelsekunde.

**Beispiel:** Sinnvolle Werte sind etwa  $N = 7$  und  $Z = 50$ .

**Programm und Beschreibung:** PH Dresden

### 3. Programme "Freizeitgestaltung"

Die Programme zur "Freizeitgestaltung" umfassen verschiedene Spiele, die zur Ausbildung des logischen Denk- und Kombinationsvermögens sowie zur Schulung eines gewissen Reaktionsverhaltens dienen. Sämtliche Programme dieser Kategorie werden auf Programmkassetten zum Kleincomputer KC 85/1 angeboten.

## R+HANOI

### Turm von Hanoi

#### Inhaltsbeschreibung:

In diesem Kombinationsspiel geht es darum, einen symbolisierten Turm von einer Position 1 zu einer Position 3 zu transportieren. Dabei darf nur eine Zwischenlagerstelle (Position 2) benutzt werden. Außerdem kann nur jeweils eine Schicht des Turmes bewegt werden, und, es sind in jedem Fall nur kleinere auf größeren Schichten abzulegen. Ziel dieses Spiels ist es, diese Aufgabe mit möglichst wenigen Zügen zu lösen.

#### Hinweise zur Programmabarbeitung

- Bei der Beantwortung der Alternativfragen erfolgt die Reaktion sofort nach Betätigen der Taste **[N]** bzw. **[J]**.
- Die Zahleneingaben bei den Verschiebeoperationen sind mit **[ENTER]** abzuschließen.
- Falsche oder unlogische Eingaben werden als solche erkannt. Es erscheint eine Fehlermeldung, die nach einigen Sekunden von selbst wieder erlischt. Das Spiel kann dann fortgesetzt werden.
- Das Programm läuft so lange, bis der Turm in vollständiger Form auf der 3. Position steht. Ein vorzeitiges Verlassen des Programms ist nur durch die **[STOP]**-Taste möglich.



R+NIM

NIM-Spiel

## Inhaltsbeschreibung:

Gegeben ist eine bestimmte Anzahl (N) von Streichhölzern. Du und der Computer nehmen abwechselnd mindestens 1, höchstens K Streichhölzer weg. Es gewinnt derjenige, der den anderen zwingt, das letzte Streichholz zu nehmen.

## Hinweise zur Programmabarbeitung:

Zuerst sind die Werte für N und K festzulegen. Dabei sind die Forderungen  $N \leq 120$  und  $K \leq N$  zu berücksichtigen. Nachdem weiterhin bestimmt wurde, wer den ersten Zug macht, kann das Spiel beginnen. Am Ende wird die Möglichkeit eines Neubeginns (eventuell mit neuem N und K) geboten.

R+MASTER

Master Mind

## Inhaltsbeschreibung:

MASTER MIND ist ein Logikspiel. Vom Computer wird eine Ziffernfolge festgelegt, die Du in möglichst wenigen Schritten zu ermitteln hast. Jede eingegebene Ziffernfolge wird vom Computer mit der zu ratenden Ziffernfolge verglichen und bewertet. Die Anzahl der  $\blacklozenge$ -Zeichen entspricht der Anzahl der Ziffern, die an der richtigen Stelle innerhalb der vom Computer vorgegebenen Ziffernfolge stehen. Die Anzahl der Ziffern, die auch in der zu ratenden Ziffernfolge auftreten, aber nicht an der richtigen Stelle stehen, ist gleich der Anzahl der angegebenen  $\blacklozenge$ -Zeichen.

## Hinweise zur Programmabarbeitung:

Du mußt folgende Werte eingeben:

Länge der Ziffernfolge: Werte zwischen 1 und 10 sind zugelassen.

erhöhte Ziffer: Eine Ziffer zwischen 1 und 9 ist einzugeben. Die zu ratende Ziffernfolge wird aus den Ziffern von 0 bis zu dem eingegebenen Wert gebildet.

Schwierigkeitsstufe: 1: Jede Ziffer des vorgegebenen Ziffernvorrautes kann in der zu ratenden Ziffernfolge nur einmal auftreten.  
2: Ziffern können mehrfach auftreten.

Danach kannst Du Deine Lösungsvorschläge eingeben. Durch Eingabe eines A anstelle eines Lösungsvorschlages wird die zu ratende Ziffernfolge ausgegeben.

Alle Eingaben sind durch Drücken der Taste **ENTER** abzuschließen.

## R+MEMORY

### MEMORY - Gedächtnistraining

#### Inhaltsbeschreibung:

Das Programm realisiert das bekannte Kartenspiel MEMORY, bei dem die Spieler versteckte Kartenpaare aus einer Anzahl verdeckter Karten aufzuspüren haben. Sieger ist derjenige, der die meisten Kartenpaare für sich verbuchen kann.

#### Hinweise zur Programmabarbeitung:

Zu Beginn werden alle 32 im Spiel vorhandenen Kartenbilder vorgestellt. Bei Schwierigkeitsstufe A werden 16 Karten ausgewählt, gemischt und verdeckt auf dem Bildschirm verteilt. Bei Schwierigkeitsstufe B werden alle 32 Karten auf zwei Bildschirmseiten verteilt, wobei der Seitenwechsel durch Eingabe der gewünschten Seitennummer erfolgt.

Mit den Kursortasten ist ein Stern auf die Karte zu positionieren, die aufgedeckt werden soll. Nach Detätigen von **ENTER** wird das Kartenbild sichtbar.

Der Pfeil vor der Spielernummer zeigt an, welcher Spieler an der Reihe ist.

## R+OTHELLO

OTHELLO - ein bekanntes Brettspiel gegen den Computer

Inhaltsbeschreibung:

Es wird mit Steinen gespielt, die durch "x" und "ø" gekennzeichnet sind. Mit "x"-Steinen spielt der Computer. Du spielst mit "ø"-Steinen.

Der Computer gibt eine Anfangsstellung vor.

Als Spielzug gibst Du jeweils ein Feld an, welches mit mindestens einem Deiner "ø"-Steine durch Überpringen eines "x"-Steines oder mehrerer "x"-Steine in horizontaler, vertikaler oder diagonaler Richtung erreicht werden kann. Dabei dürfen keine Leerfelder und keine eigenen Steine übersprungen werden. Das als Spielzug angegebene Feld muß ein Leerfeld sein. Alle Sprünge Deiner Steine, die zum angegebenen Zielfeld führen, werden nun ausgeführt, und die übersprungenen "x"-Steine des Computers werden durch Deine "ø"-Steine ersetzt. Auch das als Spielzug angegebene Feld wird mit Deinem "ø"-Stein besetzt. Dann ist der Computer am Zuge. Er spielt nach den gleichen Regeln.

Sind alle Felder besetzt oder kann ein Spieler nicht setzen (weil kein Zug möglich ist), dann ist das Spiel beendet, und der Spieler mit der größten Anzahl Steine auf der Spielfläche hat gewonnen.

Hinweise zur Programmabarbeitung:

- Bei der Frage nach dem Schwierigkeitsgrad kannst Du zwischen den Spielstufen "leicht" (1) und "schwierig" (2) wählen.
- Beantworte die Frage "Können Sie setzen" mit  N , obwohl ein Zug möglich ist; wird das Spiel vorzeitig beendet.
- Bei der Aufforderung "Ihr Zug bitte" gibst Du bei "Zeile" nur die Zeilennummer (1 bis 8) und bei "Spalte" nur die Spaltennummer (1 bis 8) ein.

# KLEINCOMPUTER robotron KC 85/1

R+HALMA

Steckhalma

## Inhaltsbeschreibung:

Entferne zunächst einen beliebigen Stein durch Angabe seiner Zeilen- und Spaltenkoordinaten aus dem Spielfeld. Überspringe dann einen Halmstein in waagerechter oder senkrechter Richtung mit einem seiner Nachbarsteine auf das entstandene freie Feld. Der übersprungene Stein wird dabei vom Programm automatisch entfernt.

Das Ziel des Spiels besteht darin, alle Steine (bis auf den letzten, der springt) aus dem Feld zu entfernen.

## Hinweise zur Programmabarbeitung:

- Tastatureingaben werden nicht mit **ENTER** abgeschlossen.

- Zum Spielen benötigte Tasten

1. für Zeilenangabe: Buchstaben A bis G
2. für Spaltenangabe: Ziffern 1 bis 7
3. für Sprungrichtungsangabe: Kursorpfeile in der untersten Tastaturzeile.

Nach Eingabe der Zeilen- und Spaltenkoordinaten kann man diese vor Ausführung des Sprungs durch Betätigen von **CONTR** **↓** noch löschen und einen neuen Stein zum Springen auswählen.

## R+FLOHSP

Das lustige Flohspringen (für 2 bis 6 Personen)

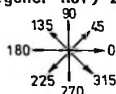
### Inhaltsbeschreibung:

Dieses Spiel beinhaltet die Idee des Springens mit farbigen Flöhen mit dem Ziel des Treffens andersfarbiger, gegnerischer Flöhe, das "Auftürmen" gefangener Flöhe und das Abliefern im eigenen Hof. Jeder Spieler hat zu Beginn 4 Flöhe zur Verfügung.

### Hinweise zur Programmabarbeitung:

Um ein gewünschtes Ziel (fremder Floh, eigener Hof) zu treffen, muß Winkel und Distanz geschätzt werden.

Ein Sprungwinkel ist entsprechend nebenstehender Skizze zu wählen. Die Distanz ist in "Flohbreiten" zu schätzen.



#### - Treffer und Aussprünge

Trifft ein Floh genau auf einen anderen, so wird dieser gefangengenommen, es entsteht ein "Turm", der vom Besitzer des obersten Flohs bewegt werden kann. Hat der Floh sein Ziel verfehlt, so verliert der Spieler im allgemeinen nur diesen einen Sprung. Springt er dagegen so, daß er außerhalb oder auf der Begrenzung des Spielfeldes landet, so gilt:

1. Ein einzelner ins Aus gesprungener Floh geht verloren und ist für immer aus dem Spiel.
2. Ein Turm wird bei Aussprung wieder im Spiel verteilt.

#### - eigener und fremder Hof

Solange sich ein Turm noch auf dem Spielfeld befindet, kann sich der Spieler der oberen gefangenen Flöhe noch nicht sicher sein. Deshalb müssen Gefangene von Zeit zu Zeit im eigenen Hof als Beute abgeliefert werden. Der Hofsprung muß dabei eines der 4 Hoffelder der eigenen Farbe treffen.

## R+ZIELE

### Ziele suchen

#### Inhaltsbeschreibung:

Im dargestellten Spielfeld, welches aus 10 mal 10 Feldern besteht, sind 4 Ziele zu suchen.

Bei jedem Versuch ist bei der Aufforderung "Zeile:" die Zeilennummer (0 bis 9) und anschließend die Spaltennummer (0 bis 9) einzugeben.

Nun wird der Abstand der noch zu suchenden Ziele (1 bis 4) vom blinkenden (bei diesem Versuch gewählten) Feld angegeben. Du hast ca. 10 Sekunden Zeit, diese Angaben für die Auswahl des nächsten Feldes im nächsten Versuch auszuwerten; danach werden sie gelöscht.

Wenn ein Ziel gefunden wurde, wird dieses mit seiner Nummer im Spielfeld eingetragen.

Nach 10 Versuchen werden die Ziele, welche Du nicht gefunden hast, vom Computer angezeigt und ins Spielfeld eingetragen.

## R+MOND

### Simulation einer Mondlandung

#### Inhaltsbeschreibung:

Ein Raumschiff befindet sich im Landeanflug über der Mondoberfläche. Infolge Ausfalls der Landeautomatik bist Du gezwungen, zur Handsteuerung überzugehen. Der Tank enthält noch einen Treibstoffvorrat, der bei geschickter Ausnutzung eine weiche Landung ermöglicht. Du mußt versuchen, eine solche Strategie der Bremsmanöver und des Treibstoffverbrauchs zu entwickeln, wie eine weiche Landung, d.h. Landung mit Geschwindigkeit Null, garantiert.

#### Hinweise zur Programmabarbeitung:

Bei Spielbeginn stellt der Computer auf Wunsch die Anfangssituation vor und gibt weitere Instruktionen. Im Verlauf der Landeoperation protokolliert der Computer auf dem Bildschirm für jeden Schritt Zeitdauer, die Höhe, Geschwindigkeit sowie den Tankinhalt und fordert dann den Spieler auf, eine Treibstoffmenge für den Bremsstoß einzugeben. Das Absinken des Raumschiffes wird in einer einfachen Grafik neben dem Protokoll angezeigt. Bei Erreichen der Mondoberfläche wertet der Computer die Landegeschwindigkeit aus.



# KLEINCOMPUTER robotron KC 85/1

## R+WORTE

### Wörter raten

#### Inhaltsbeschreibung:

Du sollst ein Wort erraten.

Das Wort wird zunächst durch waagerechte Striche angezeigt, woraus bereits die Länge des zu erratenden Wortes erkannt werden kann.

Wenn Du nun einen Buchstaben rätst (nur Taste des Buchstaben betätigen), wird dieser in das zu erratende Wort so oft eingetragen, wie er im Wort vorkommt. Anschließend ist das Wort zu raten, oder, wenn Du es noch nicht erraten kannst, die Taste **ENTER** zu betätigen.

Ist der Buchstabe nicht im Wort enthalten, dann erscheint im "Fenster" schrittweise ein Ungeheuer. Nach 8 Fehlversuchen hast Du den "Wettkampf" mit dem Ungeheuer verloren, und vom Computer wird das Wort genannt.

#### Hinweise zur Programmabarbeitung:

Bei der Eingabe des zu ratenden Wortes kannst Du das Wort beliebig korrigieren. Bist Du mit der Eingabe bzw. Korrektur des Wortes fertig, ist die Taste **ENTER** zu betätigen.

# KLEINCOMPUTER robotron KC 85/1

R+TRUMPF

Kartenspiel "Trumpf 8" oder "Mau"

## Inhaltsbeschreibung:

Es wird wie mit einem Kartenspiel, bestehend aus 52 Blatt, gespielt. Zunächst verteilt der Computer je 7 Karten an Dich und an sich selbst. Anschließend wird eine Karte aufgedeckt. Diese Karte erscheint hinter der Ausschrift "Ausgespielte Karte".

Es werden nun Deine Karten angezeigt und Du wirst aufgefordert, eine dieser Karten zu spielen. Es ist jeweils mit gleicher Farbe oder mit gleichem Kartenwert zu bedienen. Hast Du jedoch eine Karte mit dem Kartenwert "8" (Trumpf 8), dann kannst Du diese jederzeit ausspielen und anschließend eine Kartenfarbe wählen, mit der das Spiel dann fortgesetzt wird.

Kannst Du nicht bedienen, dann mußt Du solange Karten ziehen, bis Du bedienen kannst. Dies geschieht einfach dadurch, daß auf die Frage "Welche Karte spielen Sie?" keine Karte eingegeben, sondern nur die Taste **ENTER** betätigt wird.

Sind die Spielkarten aufgebraucht, erscheint die Ausschrift "Karten sind alle". Kannst Du wirklich noch nicht bedienen, dann betätige abermals die Taste **ENTER**, erhältst aber diesmal keine weitere Karte und mußt so lange aussetzen, bis Du wieder bedienen kannst.

Können beide Spieler nicht mehr bedienen, ist das Spiel zu Ende. Wer seine letzte Karte ausspielt, beendet ebenfalls das Spiel.

## Hinweise zur Programmabarbeitung:

Die Karten können kodiert eingegeben werden:

|                  |     |        |      |
|------------------|-----|--------|------|
| Dabei gilt Kreuz | - Y | Bube   | - BU |
| Pik              | - X | Dame   | - DA |
| Herz             | - C | Koenig | - KO |
| Karo             | - V |        |      |

Diese Angaben können in beliebiger Reihenfolge, mit oder ohne Leerzeichen eingegeben werden.

Z.B. Herz 7: **C** **7** **ENTER** oder **7** **C** **ENTER**

## R+PASCH

### Würfelspiel PASCH (für 1 bis 6 Mitspieler)

#### Inhaltsbeschreibung:

Ziel des Spieles ist es, im Verlauf von 15 Spielrunden so viel wie möglich Würfelkombinationen zu erreichen, die im PASCH-Protokoll aufgeführt sind. Sieger wird, wer die höchste Punktsumme erreicht.

#### Hinweise zur Programmabarbeitung:

##### - Würfeln

Nach dem ersten bzw. zweiten Wurf erscheint die Frage: "WELCHE WUERFEL BLEIBEN STEHEN?". Der Spieler wählt eine der folgenden Antworten:

N:       Nichts bleibt stehen. Der folgende Wurf wird mit allen 5 Würfeln ausgeführt.

G:       Es bleibt der gesamte Wurf stehen. Der Spieler verzichtet in dieser Runde auf weitere Würfe.

An:      Es bleiben alle Würfel, die die Augenzahl n zeigen, stehen.

$n_1 n_2 \dots$ : Für jede der 8max. 5) eingetasteten Ziffern bleibt genau ein Würfel der entsprechenden Augenzahl stehen.

Wenn ein Spieler eine Spielrunde beendet hat, wird er aufgefordert, das erzielte Wurfresultat in eine der Rubriken des PASCH-Protokolls einzutragen. Jede Rubrik darf im Verlauf der 15 Spielrunden nur einmal gewählt werden. Die Eingabe erfolgt zweistellig mit dem im PASCH-Protokoll stehenden Kurzzeichen der Rubrik, z.B. "1P" für EIN PAAR. Erfüllt ein Wurf die Bedingungen der angegebenen Rubrik nicht, so wird ein Strich (= 0 Punkte) eingetragen.

# KLEINCOMPUTER robotron KC 85/1

## - Rubriken

| Abkürzung | Bedeutung                | Bewertung  |
|-----------|--------------------------|--|
| Z1        | Zahl Eins )              | Es werden die im Wurf<br>enthaltenen Würfel<br>dieser Augenzahl<br>addiert                                   |
| Z2        | Zahl Zwei )              |  |
| Z3        | Zahl Drei )              |  |
| Z4        | Zahl Vier )              |  |
| Z5        | Zahl Fünf )              |  |
| Z6        | Zahl Sechs )             |  |
| 1P        | Ein Paar )               | Es muß die angegebene An-<br>zahl von Augenwerten gleich<br>sein, dann werden sie als<br>Punkte addiert      |
| 2P        | Zwei Paare )             |  |
| 3G        | Drei Gleiche )           |  |
| 4G        | Vier Gleiche )           |  |
| KL        | kleine Straße<br>(12345) | 15 Punkte  |
| GR        | große Straße<br>(123456) | 20 Punkte  |
| KU        | Kuchen<br>(volles Haus)  | Wurf muß 3G und 1P<br>enthalten, dann Addition<br>der Augen  |
| CH        | Chance                   | kann genutzt werden, falls<br>Wurf in keine der noch<br>freien Rubriken paßt.<br>Augenzahlen werden addiert. |
| PA        | Pasch                    | 5 gleiche Augenzahlen (egal<br>ob Einsen oder Sechsen)<br>= 50 Punkte)                                       |

## - Zwischensumme und Bonus

Für die Rubriken Z1 bis Z6 wird unter ZS ständig die aktuelle Zwischensumme ausgewiesen. Falls  $ZS \geq 63$  gilt, erhält der Spieler einen einmaligen Bonus (BO) von 50 Punkten.

## R+HOBBIT

### Hase und Wölfe - ein Strategiewettbewerb (für 1 bis 6 Spieler)

#### Inhaltsbeschreibung:

An diesem Spiel können sich 1 bis 6 Spieler beteiligen und eine Rundenzahl zwischen 1 und 10 absolvieren. Ziel des Spieles ist es, ein Häschen vor einem Rudel hungriger Wölfe zu retten, indem man es günstig auf einem Spielfeld bewegt, hinter Bäumen versteckt, Haken schlägt und so die Wölfe überlistet.

#### Hinweise zur Programmabarbeitung:

Das Spielfeld stellt ein rechteckiges "Waldstück" dar, in dem sich neben dem Häschen (weißer Stern) zu Beginn jeweils 10 Wölfe (rote Dollar-Zeichen) befinden. Jeder Spieler kann pro Runde mehrere Anfangsbilder spielen, je nachdem, wie lange er sein Häschen vor den Wölfen in Sicherheit bringt. In der Wertungsvariante 1 (Sicherheitsspiel) kann der Spieler ohne Bedenken bis zum Verlust des Hasen weiterspielen, alle Punkte bis zu diesem Zeitpunkt gehen in die Wertung. Im Gegensatz dazu zählen beim Risikospiele (Wertungsvariante 2) immer nur die Punkte abgeschlossener Bilder, falls sich der Spieler für "Aufhören" entscheidet. Spielt er weiter und "überlebt" das neue Bild nicht, so gehen alle Punkte dieser Runde verloren.

#### - Züge des Hasen

Immer, wenn rechts unten auf dem Bildschirm die Eingabeaufforderung "RICHTUNG?" erscheint, kann der Hase bewegt werden, um den Wölfen auszuweichen oder sie gegen Bäume laufen zu lassen. Als Orientierung für die Bewegungsrichtung des Hasen dient das rechts unten auf dem Bildschirm angezeigte Richtungskreuz. Der Hase darf nicht gegen Bäume, Mauern und Wölfe laufen. Für den Zug des Hasen besteht eine Zeitbegrenzung von ca. 5 Sekunden.

Alle Wölfe ziehen nach dem Hasenzug um ein Feld entsprechend ihrer Strategie weiter, "sehen" aber Bäume und Mauern nicht. Mitunter ist der beste Zug des Hasen, auf der Stelle stehen-zubleiben. Dies wird durch Betätigen der Taste "S" oder durch Überschreiten der Zeitbegrenzung erreicht. Der Hase darf höchstens 2mal hintereinander stehenbleiben, sonst schläft er für dieses Bild ein und die Wölfe haben freie Bahn. Befindet sich ein Wolf direkt neben dem Hasen, so sollte der Hase vorsichtshalber hakenschlagend fliehen (d.h. mit Zügen in diagonaler Richtung). Nach dem dritten Zug in gerader Richtung ist sonst der Wolf der Schnellere.

## - Bildwechsel

Sind alle Wölfe eines Bildes besiegt, so erscheint für den glücklichen Spieler (eventuell auf Wahl in Wertungsvariante 2) ein neues Bild. Jedes neue Bild einer Runde hat weniger Bäume bei gleicher Anzahl Wölfe.

Hat der Spieler seinen Hasen verloren, so ist der nächste an der Reihe, für ihn wird das 1. Bild angezeigt.

# KLEINCOMPUTER robotron KC 85/1

R+SKEET

SKEET - Wurftaubenschießen

## Inhaltsbeschreibung:

SKEET ist ein Geschicklichkeitsspiel, das Reaktionsgeschwindigkeit und Orientierungsvermögen erfordert und schult.

Der Computer läßt nacheinander verschiedene Wurftauben aus zufällig gewählten Richtungen über den Bildschirm fliegen. Gleichzeitig wird am unteren Bildrand der Standort des Schützen angezeigt. Durch Betätigen der Leer-Taste (SPACE) wird ein "Schuß" ausgelöst.

Du mußt nun versuchen, den Zeitpunkt des Abschusses so zu wählen, daß die Wurftauben getroffen werden. Die Treffer werden optisch signalisiert und in einer Trefferanzeige mitgezählt. Nach 50 Schuß gibt der Computer eine Auswertung auf dem Bildschirm aus.

## Hinweise zur Programmabarbeitung:

Der Computer fragt nach der gewünschten Schwierigkeitsstufe. Dabei bedeuten:

- 1 - "Schütze" bleibt immer am selben Ort.
- 2 - Der Standort des "Schützen" wird vor jedem Schuß zufällig eingestellt (innerhalb eines gewissen Bereiches am unteren Bildrand).

Auf jede Wurftaube darf nur ein "Schuß" abgegeben werden, anderenfalls gibt es keine Möglichkeit, die nächste Taube zu treffen.

## R+AUTOCR

### AUTOCROSS - Reaktionsprogramm mit Spielhebel-Steuerung

#### Inhaltsbeschreibung:

AUTOCR ist ein Reaktionspiel. Mit dem Spielhebel ist ein Auto über eine beliebige Hindernisstrecke zu steuern. Die Fahrgeschwindigkeit ist in 3 Stufen wählbar. Die Fahrzeit und die aufgetretenen Karambolagen werden ermittelt und bewertet.

#### Hinweise zur Programmabarbeitung:

Die Fahrstrecken bzw. die Hindernisse auf der Strecke werden entweder vom Programm vorgegeben (Standardstrecke) oder beliebig aufgebaut. Der Aufbau erfolgt über Spielhebel. Mit dem Spielhebel wird der Cursor bewegt. Ist gleichzeitig die Aktionstaste am Spielhebel gedrückt, wird die aktuelle Kursorposition als Begrenzungsfeld eingetragen. Ist gleichzeitig die Taste **[SPACE]** auf der Tastatur gedrückt, wird ein Leerzeichen auf die aktuelle Kursorposition eingetragen. Mit der Taste **[ENTER]** wird der Aufbau der Fahrstrecke abgeschlossen.

Nachdem der Spieler seine Startbereitschaft über die Taste **[ENTER]** signalisiert hat, beginnt die Zeitmessung für die Fahrt vom Start- zum Zielpunkt. Aus Fahrzeit, Zahl der Karambolagen und gewählter Geschwindigkeit wird je Fahrt eine Punktzahl errechnet und ausgegeben. Für mehrere Fahrten über dieselbe Strecke, d.h. bei unverändertem Streckenaufbau, wird die erreichte Punktzahl mit dem bisherigen Spitzenwert verglichen.



## R+SLALOM

### Abfahrtslauf - ein Geschicklichkeitspiel

#### Inhaltsbeschreibung:

Die Figur des Skiläufers kann mittels Kursorsteuertasten  und  nach rechts bzw. links verschoben werden.

Es hängt vom Geschick des Spielers ab, in der jeweiligen Spielsituation Zeitpunkt und Dauer der Tastenbetätigung so zu wählen, daß Fähnchentore passiert werden oder Hindernissen ausgewichen wird.

Ein Lauf besteht aus 3 Runden, wobei zum Erreichen der folgenden Runde stets ein Mindestwert an Punkten nötig ist. Sind alle drei Runden absolviert, erscheint eine Endauswertung.

#### Hinweise zur Programmabarbeitung:

- Runde 1:** Jedes passierte Tor zählt 10 Punkte mindestens 500 Punkte sind notwendig, um die 2. Runde zu erreichen. Jeder Zusammenstoß mit Hindernissen kostet 50 Punkte. Das Auslassen eines Töres bringt keine Punktabzüge, aber das nächste Tor kann nicht mehr ausgelassen werden (Zaun entsteht).
- Runde 2:** Fähnchen müssen vom Skiläufer abwechselnd rechts und links auf der Fahnenstangenseite passiert werden. Runde endet, wenn die Fehlergutschrift aus Runde 1 verbraucht ist oder nach 200 Fähnchen der erste Fehler passiert.
- Runde 3:** Ablauf ähnlich Runde 1 Auslassen von Toren nicht möglich (Zaun erscheint immer). Der erste Fehler beendet die Runde sofort.

#### 4. STARTHILFE ZUR BASIC-PROGRAMMIERUNG

Die folgende kurze Einführung in die Anfänge der BASIC-Programmierung soll Dich befähigen, ohne besondere Vorkenntnisse den KC 85/1 für die Lösung erster eigener kleiner Aufgaben zu nutzen sowie das Verständnis für die angebotenen Beispielprogramme fördern. Dabei wird nur auf einige der wichtigsten Anweisungen und Kommandos eingegangen. Was BASIC noch alles kann, darüber kannst Du Dich z.B. in der BASIC-Obersicht (Faltkarte) informieren.

Bevor Du Deine praktischen BASIC-Übungen beginnst, solltest Du Dir einige Grundkenntnisse über den Umgang mit dem KC 85/1 verschafft haben. Dazu nutzt Du am besten die entsprechenden Hinweisblätter.

Was ist nun ein BASIC-Programm, und wie schreibt man es?

Das BASIC-Programm ist eine Folge von numerischen BASIC-(Programm-)Zeilen, die Anweisungen enthalten und in aufsteigender Folge der Zeilennummern nacheinander ausgeführt werden.

Die BASIC-Zeile enthält eine oder mehrere BASIC-Anweisungen, die durch Doppelpunkt voneinander getrennt werden. Eine BASIC-Zeile kann sich über fast zwei Zeilen des Bildschirms erstrecken (max. 72 Zeichen). Sie kann im Kommandomodus (kenntlich durch OK auf dem Bildschirm) einzeln eingegeben werden und wird entsprechend ihrer Zeilennummer in das Programm einsortiert.

Die BASIC-Zeilenummer muß im Bereich von 1 bis 65527 liegen.

Die BASIC-Anweisung bestimmt, was der Computer zum entsprechenden Zeitpunkt ausführen soll. Sie darf natürlich nicht länger als eine BASIC-Zeile sein. Wird sie im Kommandomodus ohne Zeilennummer eingegeben, wird sie sofort ausgeführt (und geht nicht in das BASIC-Programm ein!).

Das BASIC-Kommando dient hauptsächlich der Manipulation des BASIC-Programmes (z.B. Starten, Anzeigen, Ändern, ...) und wird ebenfalls ohne Zeilennummer im Kommandomodus eingegeben. Teilweise existieren dafür auch spezielle Tasten (RUN, LIST). Bevor wir uns mit BASIC-Anweisungen beschäftigen, machen wir uns mit einigen Kommandos vertraut. Wir benötigen sie für unsere praktische Arbeit mit dem Computer.

#### WICHTIGE KOMMANDOS

Wollen wir ein Programm eingeben, egal ob über die Tastatur oder durch Einlesen von MB-Kassette, sorgen wir mit

**NEW** (Löschen Programmspeicher)

zunächst dafür, daß ein eventuell noch vorhandenes BASIC-Programm gelöscht und der Speicher freigemacht wird. (Zur Erinnerung: Eingaben von der Tastatur werden mit der ENTER-Taste abgeschlossen!)

Wir können nun unser Programm zeilenweise eingeben. Die vorangestellte Zeilennummer bestimmt die Einordnung in das BASIC-Programm und darf nicht vergessen werden. Leichter geht es oft, wenn mit dem Kommando

AUTO (automatische Zeilennummerierung)

die Zeilennummern (in 10er Schritten, mit 10 beginnend) fortlaufend erzeugt werden. Zur Rückkehr in den Kommandomodus ist zum Schluß die STOP-Taste zu drücken.

Wollen wir jedoch unser Programm von Kassette einlesen, geben wir nach NEW z.B.

CLOAD "R+FLOT" (Laden BASIC-Programm)

ein und können auf diese Weise ein unter dem Namen R+FLOT aufgezeichnetes Programm in den Computer zurückholen. (Beachte die gesonderten Hinweise zur Arbeit mit dem Kassettenrekorder!)

Wenn wir unser Programm, bevor wir seine Ausführung starten, noch einmal ansehen wollen, drücken wir die Taste LIST

LIST (Anzeigen BASIC-Programm)

und bekommen die ersten zehn BASIC-Zeilen angezeigt. Nach Betätigung der ENTER-Taste sehen wir die nächsten zehn Zeilen usw. bis das Programm endet bzw. die STOP-Taste gedrückt wird.

Nun endlich wollen wir unser Programm ausprobieren.

Dazu starten wir es ab seiner niedrigsten BASIC-Zeilenummer mit

RUN (Starten BASIC-Programm)

Mit der STOP-Taste können wir seine Abarbeitung bei Bedarf auch wieder abbrechen, z.B. um noch etwas zu korrigieren.

Durch Eingabe im Kommandomodus können nun, entsprechend ihrer Zeilennummer, BASIC-Zeilen ergänzt und/oder vorhandene durch neue ersetzt werden. Will man jedoch nur eine kleine Veränderung einer bereits vorhandenen BASIC-Zeile vornehmen, z.B. in der Zeile mit der Nummer 120, so genügt

EDIT 120 (Ändern von Programmzeilen)

um die Zeile anzeigen zu lassen und ebenso korrigieren zu können wie bei der Ersteingabe (Verwendung Kursortasten, Einfügen/Löschen von Zeichen). Wichtig ist, daß die Übernahme der geänderten BASIC-Zeile in den Programmspeicher nur erfolgt, wenn sie mit der ENTER-Taste abgeschlossen wird.

Soll unser Programm zum Schluß aufbewahrt werden, müssen wir es auf Kassette aufzeichnen (sichern) z.B.:

CSAVE "PROG1" (Speichern BASIC-Programm)

Dabei müssen wir einen bis zu 8 Zeichen langen Programmnamen (hier: PROG1) vergeben, der beim Wiedereinlesen erneut anzugeben ist (vgl. CLOAD). Nach Aufzeichnungsende erscheint auf dem Bildschirm die Anfrage "VERIFY (Y/N)?". Sie bedeutet, daß der Bediener entscheiden muß, ob das aufgezeichnete Programm zur Kontrolle noch einmal eingelesen (Antwort: Taste **Y**) werden soll oder nicht (Antwort: Taste **N**). Im ersten Fall ist die Aufforderung "REWIND:" durch Rückspulen der Kassette auf den Anfang der Aufzeichnung, Einschalten der Wiedergabe und Betätigung der **ENTER**-Taste bei Ertönen des Vortones zu beantworten.

Weiteres "Werkzeug" in Form von Kommandos benötigen wir zunächst nicht. Wir beginnen jetzt mit der eigentlichen BASIC-Programmierung.

## EINIGE BASIC-ANWEISUNGEN

Bei der Formulierung von Anweisungen achte bitte auf den Typ der Daten, die zu verarbeiten sind. Es gibt Zahlen und Zeichenketten (Aneinanderreihung von Zeichen des Zeichensatzes). Wollen wir Daten auf den Bildschirm (24 x 40 Zeichen) ausgeben, so schreiben wir z.B.

```
100 PRINT 3*77          (Ausgabeanweisung)
```

und erhalten das Ergebnis der Multiplikation (die Zahl 231) am Anfang der nächsten freien Zeile ausgegeben.

Für die Zahlendarstellung ist zu beachten, daß

- stets eine Vorzeichenstelle (Leerzeichen falls positiv) und ein nachfolgendes Leerzeichen ausgegeben werden,
- Zahlen, die betragsmäßig größer 999999 oder kleiner 0.01 sind, automatisch im Gleitkommaformat ausgegeben werden (z.B. 1.2E+7 statt 12000000),
- Zahlen betragsmäßig nicht größer als 1.7E+38 sein dürfen und
- grundsätzlich der Dezimalpunkt statt des Dezimalkommas zu verwenden ist.

Analog können wir eine Zeichenkette ausgeben:

```
220 PRINT "ABC"
```

Dabei ist wichtig, daß die Anführungszeichen lediglich als Begrenzer verwendet werden, aber nicht zur Zeichenkette gehören und demzufolge auch nicht ausgegeben werden.

Zeichenketten werden ohne vorangestellte oder nachfolgende Leerzeichen ausgegeben.

Selbstverständlich können mit einer einzigen Ausgabeanweisung auch mehrere Daten ausgegeben werden. Dann ist wichtig, mit welchem Trennzeichen (Komma oder Semikolon) die einzelnen Ausgabelemente voneinander getrennt sind, z.B. bewirkt

```
100 PRINT 1, 2, 3
```

die Ausgabe der Zahlen 1, 2 und 3 linksbündig in einem jeweils 13 Zeichen breiten Feld auf einer Bildschirmzelle (Tabellenausgabe). Bei Verwendung des Semikolons in

```
760 PRINT 2048;"byte"
```

erhält man demgegenüber eine unmittelbar aufeinanderfolgende Ausgabe der beiden Elemente.

Benutzen wir PRINT ohne weitere Angaben, wird eine Leerzeile ausgegeben. Obigens kannst Du zur Erleichterung der Tastatureingabe statt des Schlüsselwortes PRINT einfach ? eingeben!

Hin und wieder wird der Wunsch bestehen, vor der Ausführung weiterer Ausgaben den Bildschirm zu löschen und danach die oberste Zeile als aktuelle Ausgabezeile zu haben. Das geschieht durch die Anweisung

20 CLS (Löschen Bildschirm)

und wird auch häufig in Form einer sofort ausführbaren Anweisung im Kommandomodus genutzt.

Nun wollen wir natürlich nicht nur mit konstanten Werten operieren, sondern uns beliebige Eingabedaten oder Zwischenergebnisse unter einem selbstgewählten Namen speichern, d.h. mit Variablen arbeiten. Dabei haben wir wieder die beiden Datentypen zu unterscheiden: numerische und Zeichenkettenvariable. Die Festlegung trifft der Programmierer mit der Namensvergabe. Variablennamen müssen mit einem Buchstaben beginnen und nur die ersten beiden Zeichen werden zur Unterscheidung benutzt. Im Unterschied zu numerischen Variablen besitzen alle Zeichenkettenvariablen ein Dollarzeichen als letztes Zeichen ihres Namens (z.B. ist X1 eine numerische, T\$ aber eine Zeichenkettenvariable).

Mit der Anweisung

60 LET X1=10  
oder 60 X1=10 (Wertzuweisung)

wird der (numerischen) Variablen X1 der Wert 10 als aktueller Wert zugewiesen. Diese Zuweisungsoperation darf nicht mit einer mathematischen Gleichung verwechselt werden! Es gilt die Regel: "Der Wert des rechtsstehenden Ausdrucks wird berechnet und anschließend der linksstehenden Variablen als aktueller Wert zugewiesen."

Der "Ausdruck" kann nun sowohl eine Konstante als auch eine komplizierte Formel sein, z.B.:

140 X1=10+3\*SIN(X1/4)\*COS(2\*Y)

Wie das Beispiel zeigt, darf der Name der linksstehenden Variablen im Ausdruck auf der rechten Seite auftreten, was die oben angegebene Regel ohne weiteres erlaubt. Wichtig ist, daß auf beiden Seiten der Wertzuweisung derselbe Datentyp stehen muß!

Weiter entnehmen wir obigem Beispiel, daß die BASIC-Operationszeichen von der üblichen Schulschreibweise etwas abweichen (\* für Multiplikation, / für Division, ^ für Potenzierung) und, daß einige Standardfunktionen zur Verfügung stehen (z.B. sin(x), cos(x), ...).

Für Zeichenketten gibt es nur einen Operator (+), der die Verkettung (Aneinanderreihung) von Zeichenketten bewirkt. Z.B. würde die Anweisungsfolge

70 T1\$="PALAST"  
80 PRINT T1\$+"CAFE"

die Zeichenkette PALASTCAFE auf dem Bildschirm erzeugen.

Auch für Zeichenketten sind in BASIC eine Reihe von Funktionen benutzbar, die die Ermittlung der Länge von Zeichenketten (Zeichenzahl), die Wiederholung von Zeichenketten, das Herausschneiden von Teilzeichenketten usw. ausführen.

Wollen wir Variablen, während das Programm läuft, über die Tastatur Werte zuweisen, so geschieht dies beispielsweise durch

```
110 INPUT "ZAHL=";Z (Eingabeanweisung)
```

Bei Ausführung dieser Anweisung schreibt der KC 85/1 den in Anführungszeichen stehenden Aufforderungstext auf dem Bildschirm aus und wartet auf einen Eingabeabschluß mittels **ENTER**-Taste. Wurde eine Zahl eingegeben, besitzt die Variable Z anschließend diesen Wert. Anderenfalls (Leereingabe, d.h. sofortige **ENTER**-Betätigung) bleibt der Wert von Z unverändert. Es können mit einer INPUT-Anweisung gleichzeitig mehreren Variablen Werte zugewiesen werden. Die Variablen und die Eingabewerte sind dann in derselben Reihenfolge und durch Komma getrennt aufzuführen, z.B. wird

```
740 INPUT "P,Q?"; P,Q
```

nach der Eingabe von 3,-4.1 bewirken, daß die Werte P=3 und Q=-4.1 angenommen werden. Ebenso ist die Werteingabe über Tastatur auch für Zeichenkettenvariable möglich.

Mit den bis jetzt vorgestellten Anweisungen können sogenannte "Geradeausprogramme" geschrieben werden, also keine Verzweigungen (Sprünge) im Programmablauf vorgeschrieben werden. Oft wird für die Lösung eines Problems jedoch eine solche Möglichkeit benötigt. Sie existiert in BASIC einmal in der Form

```
300 GOTO 210 (unbedingte Verzweigung)
```

und würde im Beispiel zwingend die Programmfortsetzung auf der BASIC-Zeile mit der Nummer 210 bewirken. Zum anderen kann die Programmverzweigung von der Erfüllung einer Bedingung abhängig gemacht werden, z.B.

```
410 IF A<5 THEN 340 (bedingte Verzweigung)
```

Hier wird das Programm mit Zeile 340 fortgesetzt, wenn die Variable A aktuell einen Wert hat, der kleiner 5 ist. Anderenfalls geschieht die Programmfortsetzung in der nächstfolgenden BASIC-Zeile.

Es ist möglich, hinter THEN auch andere Anweisungen anzugeben (z.B. eine PRINT-Anweisung), die immer nur dann ausgeführt würden, wenn die unter IF angegebene Bedingung erfüllt ist. Eine erweiterte Form

```
600 IF A<0 THEN PRINT "A<0":ELSE PRINT "A>=0"
```

gestattet die ausdrückliche Angabe von Anweisungen sowohl für den Fall der Erfüllung (hinter THEN) als auch der Nichterfüllung (hinter ELSE) der Bedingung.

Abschließend schauen wir uns eine weitere wichtige Programmsteueranweisung an, nämlich die Wiederholungs- oder Schleifenanweisung. Nehmen wir an, wir wollen eine kleine "Zahlentafel" ausgeben. Wir schreiben

```
71Ø FOR I=2 TO 4Ø STEP 2
72Ø PRINT I, I*I,1/I           (Wiederholungsanweisung)
73Ø NEXT I
```

was bedeutet, daß die zwischen FOR... und NEXT liegenden Anweisungen (bei uns nur eine PRINT-Anweisung) solange ausgeführt werden, bis der Wert der Laufvariablen (hier I) den vorgegebenen Endwert (hier 4Ø) überschritten hat. Dabei wird, mit dem Anfangswert für I (hier 2) beginnend, nach jedem Durchlauf die hinter STEP angegebene Schrittweite (hier 2) zum aktuellen Wert der Laufvariablen hinzuaddiert. Beträgt die Schrittweite 1, kann die STEP-Angabe entfallen. In unserem Beispiel erhalten wir also eine Tabelle mit 2Ø Zeilen ( $I = 2, 4, \dots, 4Ø$ ). Anschließend wird das BASIC-Programm mit der auf NEXT folgenden BASIC-Anweisung fortgesetzt.

Beachtet werden muß, daß der Eintritt in eine Wiederholungsanweisung nur über FOR... erfolgen darf, also kein "Hineinspringen" (z.B. mit GOTO) stattfindet. Es ist möglich, mehrere Wiederholungsanweisungen ineinander "zu schachteln". Dabei muß die zuletzt eröffnete Schleife zuerst geschlossen werden (NEXT!) usw.

Spätestens jetzt solltest Du Dich an das Probieren kleiner Beispielprogramme bzw. die Lösung selbstgestellter Aufgaben heranzuwagen. Denn: Übung macht den Meister! Programmfehler, die Du als Anfänger sicher hin und wieder machen wirst, können den KC 85/1 nicht zerstören und sind korrigierbar (EDIT-Kommando!).

Viel Spaß und Erfolg bei der BASIC-Programmierung!



## 5. Einfache Funktionswerte

**Inhalt:** Berechnung von Funktionswerten einer vorgegebenen Funktion.

Das Argument X wird vom Programm angefordert und muß über die Tastatur eingegeben werden.

**Programm:**

```
1Ø INPUT "Argument X ="; X
2Ø Y = 15*X - 3      | Funktionswert Y
3Ø PRINT Y
4Ø GOTO 1Ø
```

**Hinweise:**

- Bei der Zahleneingabe muß ein Dezimalpunkt (keine Komma) gesetzt werden.

- Das Programm wird durch **STOP** beendet.

**Varianten:** Durch Änderung der Zeile 20 lassen sich beliebige andere Funktionswerte berechnen.

Dabei können die Grundrechenarten (+, -, \*, /) sowie die vorhandenen Standardfunktionen, z.B.

SQR(X) - Quadratwurzel von X  
SIN(X) - Sinusfunktion  
EXP(X) - Exponentialfunktion,  
verwendet werden.

## Benchmark-Test

- Inhalt:** Bestimmung der Rechenzeit einzelner Operationen bzw. Standardfunktionen.
- Programmi**
- ```
1Ø PRINT "START"; BEEP
2Ø FOR I = 1 TO 2ØØØ
3Ø A = 5 * 3           :| Trivialoperation
4Ø NEXT I
5Ø BEEP; PRINT "STOP", A
6Ø END
```
- Hinweise:**
- Die Laufzeit des Programmes kann mit einer Armbanduhr ermittelt werden. Die gestoppte Zeit wird dann durch 2000 dividiert.
- Varianten:**
- Durch Weglassen der Zeile 30 sollte auch die Zeit ermittelt werden, die nur durch die FOR...NEXT-Schleife benötigt wird.
  - In Zeile 30 können verschiedene Operationen programmiert werden. Z.B. die Grundrechenarten +, -, \*, / oder auch Standardfunktionen wie SIN, EXP, ABS usw.
  - Man beachte die sehr lange Rechenzeit für die Potenzoperation  $\wedge$ .  
(Aufgrund der Rechenzeit und der relativen Ungenauigkeit dieser Funktion sollte man sie in Programmen möglichst vermeiden.)

## Zeichensatz

**Inhalt:** Anzeigen aller darstellbaren Zeichen (Ziffern, Buchstaben, Sonderzeichen, Grafikzeichen) des Computers auf dem Bildschirm.

**Programm:**

```
1Ø WINDOW: CLS
2Ø FOR I = 32 TO 127
3Ø PRINT CHR$(I);
4Ø NEXT I
5Ø PRINT
6Ø FOR I = 128 TO 255
7Ø PRINT CHR$(I);
8Ø NEXT I
9Ø END
```

**Hinweis:** Das Semikolon jeweils am Ende der Anweisungen 30 und 70 darf nicht vergessen werden.

**Varianten:**

- Durch Änderung der Zeilen 30 und 70 kann die Bildgestaltung variiert werden.  
z.B.  
3Ø PRINT CHR\$(I);" \*";
- Auch das Einfügen von Farbanweisungen (INK, PAPER) ist möglich.

## Elektronischer Würfel

**Inhalt:** Nutzung des Zufallsgenerators (RND-Funktion) von BASIC zur Erzeugung zufälliger, natürlicher Zahlen im Bereich von 1 bis 6.

**Programm:**

```
10 PRINT "KC 85/1 WUERFELT NACH > ENTER <!"
20 PRINT: WURF = 0
30 INPUT " "; X$
40 WURF = WURF + 1
50 Zahl = INT(6*RND(1)) + 1
60 PRINT WURF; "-TER WURF:"; ZAHL
70 GOTO 30
```

**Hinweise:** Zeile 30 dient lediglich der Wurfauslösung mittels ENTER-Taste.

Der Zufallsgenerator liefert einen Wert zwischen

$$0 < \text{RND}(1) < 1,$$

aus dem durch Multiplikation mit 6 und Abschneiden des gebrochenen Teils (Funktion INT) eine ganze Zahl zwischen 0 und 5 gewonnen wird.

**Varianten:**

1. Würfelt man mit einem Zwölfflächner (Dodekaeder) sind Zufallszahlen zwischen 1 und 12 erforderlich:  
50 ZAHL = INT(12\*RND(1)) + 1

2. Ergänzung einer Wurfstatistik:

```
5 WINDOW:CLS:DIM HZ(6)
12 PRINT
14 PRINT "WURF-STATISTIK: ZAHL HAEUFIGKEIT"
16 FOR I = 1 TO 6: PRINT AT(I+3,19); I
18 PRINTAT (I+3,27); 0
19 NEXT I
25 WINDOW 11,23,439
62 HZ(ZAHL) = HZ(ZAHL) + 1
64 PRINTAT (ZAHL+3,27); HZ(ZAHL)
```

**Achtung:** Nach **STOP** ist das Rücksetzen des Ausgabebereiches mittels WINDOW erforderlich!

## Übungsprogramm Addition

- Inhalt:** Es werden zufällig zwei Zahlen zwischen 0 und 50 vom Rechner gebildet. Du wirst aufgefordert, die Summe einzugeben. Anschließend wird geprüft, ob der eingegebene Wert richtig ist.
- Programm:**
- ```
10 Z1= INT(50*RND(1)+.5)
20 Z2= INT(50*RND(1)+.5)
30 PRINT "1. Summand: "; Z1
40 PRINT "2. Summand: "; Z2
50 INPUT "Summe ? "; S
60 IF Z1 + Z2 = S THEN PRINT "richtig":
ELSE PRINT "falsch"
70 PRINT: GOTO 10
```
- Hinweis:** Mit den Anweisungen in den Zeilen 10 und 20 werden die Summanden ausgelost. RND(1) liefert einen zufälligen Wert zwischen 0 und 1. INT(argument) hat als Funktionswert die größte ganze Zahl, die im argument enthalten ist. Durch Drücken der **STOP**-Taste kann das Programm beendet werden.
- Varianten:**
- Wenn man die 50 in den Zeilen 10 und 20 durch einen anderen Wert ersetzt, wird die Grenze des Bereiches, aus dem die Zahlen ausgewählt werden, verändert.
  - Laß den Rechner die Anzahl richtiger und falscher Antworten zählen. Füge die Zeilen 5 und 65 ein und ändere Zeile 60:  
5 R=0: F=0  
60 IF Z1+Z2 = S THEN PRINT "richtig": R=R+1  
ELSE PRINT "falsch": F=F+1  
65 PRINT "Bisher";R;"richtige,";F;"falsche Eingaben."
  - Übungsprogramme für Subtraktion und Multiplikation können ähnlich erstellt werden. Etwas schwieriger ist es bei der Division, wenn das Ergebnis ganzzahlig sein soll. Vielleicht überlegst Du Dir einmal allein, wie ein Divisionsübungsprogramm aussehen müßte.

## Raten einer Zahl

**Inhalt:** Der Rechner lost eine Zahl zwischen 0 und 1023 aus, die Du raten sollst. Wenn Du dazu aufgefordert wirst, gib eine Zahl ein, von der Du denkst, daß sie richtig ist. Drücke nach der Eingabe die ENTER-Taste.

**Programm:**

```
10 MA = 1023
20 Z = INT(MA*RND(1)+5.5)
30 PRINT
40 INPUT "Rate die Zahl!"; R
50 IF R<Z THEN PRINT "zu klein": GOTO 30
60 IF R>Z THEN PRINT "zu gross": GOTO 30
70 PRINT "richtig"
```

**Hinweis:** RND(1) liefert einen zufälligen Wert zwischen 0 und 1. INT(**argument**) hat als Funktionswert die größte ganze Zahl, die im **argument** enthalten ist.

**Varianten:**

- Die obere Grenze des Bereiches, aus dem Zahlen ausgewählt werden, wird verändert, wenn man MA in Zeile 10 einen anderen Wert zuweist.
- Die Anzahl der Rateversuche wird gezählt, wenn man folgende Zeilen in das Programm einfügt:  
5 V = 0  
45 V = V+1  
80 PRINT: PRINT V;"Versuche"

## Funktionsstabellierung

**Inhalt:** Zwei Funktionen Y1 und Y2 werden im Intervall [C, D] tabelliert. Für die Tabellierungsschrittweite S wird zunächst S = 1 angenommen.

**Programm:**

```
10 INPUT "KLEINSTER WERT = "; C
20 INPUT "GROESSTER WERT = "; D
30 S = 1
40 PRINT "X", "Y1 (X)", "Y2 (X)"
50 PRINT "-----"
60 FOR X = C TO D STEP S
70 Y1 = X * X      ! Quadrat
80 Y2 = SQR(X)    ! SQR = Wurzelfunktion
90 PRINT X, Y1, Y2
100 NEXT X
110 END
```

- Varianten:**
- Die Funktionen Y1 und Y2 (in den Zeilen 70 und 80) können variiert werden.
  - Eine bessere Bildgestaltung erreicht man durch einfügen von

```
35 WINDOW:CLS
55 WINDOW 3,23,0,39
```
  - Die Schrittweite S in Zeile 30 kann geändert oder auch über INPUT eingegeben werden.
  - Bei bestimmten Schrittweiten S wird der Wert  $X = D$  nicht mehr erreicht. Das hängt mit der rechnerinternen Zahlendarstellung zusammen. Wird zu D eine kleine Zahl addiert (z.B.  $10^{-5} * D$ ), so wird diesem Effekt entgegengewirkt.

## Kurvendarstellung (mit Print-Plot)

**Inhalt:** Der Verlauf der Funktion  $y = \sin(x + \pi)$  wird für  $x$ -Werte zwischen 0 und  $2\pi$  mit einer Schrittweite von  $\frac{\pi}{6}$  dargestellt. Die  $x$ -Achse verläuft vertikal, die  $y$ -Achse horizontal.

**Programm:**

```
10 MI = -1: MA = 1
20 WINDOW: CLS
30 PRINT "Y = SIN(X + PI)"
40 FOR X = 0 TO 2*PI STEP PI/6
50 Y = SIN(X+PI)
60 PRINT X;
70 PRINT TAB(14+(Y-MI)/(MA+MI)*24); "*"
80 NEXT X
```

**Hinweis:** Den Variablen MI und MA werden minimaler bzw. maximaler Funktionswert, die sich für die  $x$ -Werte des darzustellenden Intervalls ergeben, zugewiesen. Entsprechend dem berechneten Funktionswert wird ein \* in eine der Spalten 14 bis 38 gesetzt.

**Varianten:**

- Die spaltengerechte Ausgabe von minimalem und maximalem Funktionswert in einer Kopfzeile wird durch folgende Anweisungen erreicht.  
32 PRINT TAB(14); MI;  
34 PRINT TAB(39-LEN(STR\$(MA))); MA
- Die Ausgabe der Bilder anderer Funktionen erfordert Änderungen in den Zeilen 10 (MI-, MA-Zuweisung), 30 (Überschrift), 40 (x-Intervall) und 50 (Funktionswertberechnung)



## Quadratische Gleichung

**Inhalt:** Berechnung der Lösungen von  $x^2+px+q = 0$  nach Vorgabe von p und q:

$$x_1 = -p/2 + \sqrt{p^2/4 - q}$$

$$x_2 = -p/2 - \sqrt{p^2/4 - q}$$

**Programm:**

```
1Ø WINDOW: CLS
2Ø PRINT "LOESUNG VON X^2+P*X+Q=Ø": PRINT
3Ø INPUT "P ="; P
4Ø INPUT "Q ="; Q : PRINT
5Ø D = P * P/4 - Q
6Ø IF D < Ø THEN PRINT "KEINE LOESUNG";
PRINT:END
7Ø PRINT "X1="; - P/2 + SQR(D)
8Ø PRINT "X2="; - P/2 - SQR(D)
9Ø PRINT: END
```

**Hinweis:** PRINT alleinstehend bedeutet Ausgabe einer Leerzeile auf dem Bildschirm und kann auch weggelassen werden.

**Varianten:** 1. Zusammenfassung der Eingabe von P und Q in einer INPUT-Anweisung z.B.:

```
3Ø INPUT "P,Q ?"; P, Q
4Ø PRINT
```

2. HINWEIS auf zulässigen Q-Wert nach Eingabe von P, z.B.

```
35 PRINT: PRINT "Q KLEINER ODER GLEICH";
P*P/4; "WAEHLINI": PRINT
```

## Zeichenkettenoperationen

**Inhalt:** Eine Zeichenkette (Folge aus Buchstaben, Ziffern, Sonderzeichen, ...) wird eingegeben. Mit Hilfe der Zeichenkettenfunktionen LEN und STRING\$ wird sie entsprechend ihrer Länge unterstrichen.

**Programm:**

```
10 INPUT "ZEICHENKETTE:";Z$
20 L = LEN(Z$)
30 PRINT: PRINT
40 PRINT"Die eingegebene Zeichenkette"
50 PRINT
60 PRINT TAB(8);Z$
70 PRINT TAB(8);STRING$(L, "=")
80 PRINT
90 PRINT"besteht aus ";L;"Zeichen."
100 PRINT
110 PRINT"-----"
120 END
```

- Varianten:**
- Die Zeichenkette kann vor der Ausgabe auf maximal 30 Zeichen (linksbündig) begrenzt werden durch

```
25 IF L > 30 THEN Z$=LEFT$(Z$,30):L=30
```
  - Durch Einfügen einer CLS-Anweisung kann die Bildgestaltung verbessert werden.

## Zeichenkettensortierung

**Inhalt:** Abspeichern der über Tastatur eingegebenen Zeichenketten in einem Feld, sortieren dieser Zeichenketten und Ausgabe der sortierten Zeichenketten auf dem Bildschirm.

```
1Ø WINDOW:CLS
2Ø INPUT "Anzahl der Zeichenketten:";N
3Ø PRINT
4Ø DIM ZKØ(N)
5Ø FOR I=1 TO N
6Ø PRINT"ZK";I;: INPUT": ";ZKØ(I)
7Ø NEXT
8Ø PRINT
9Ø FOR J=1 TO N-1
1ØØ FOR I=1 TO N-1
11Ø IF ZKØ(I+1) < ZKØ(I) THEN XØ=ZKØ(I);
ZKØ(I)=ZKØ(I+1) ; ZKØ(I+1)=XØ
12Ø NEXT I,J
13Ø FOR I=1 TO N
14Ø PRINT ZKØ(I)
15Ø NEXT
```

**Hinweis:** Das Semikolon nach dem I in Zeile 6Ø bewirkt, daß die INPUT-Anweisung auf der gleichen Bildschirmzeile abgearbeitet wird.

**Varianten:**

- Wieviel Zeichenketten können in dem mit DIM ZKØ(N) dimensionierten Feld gespeichert werden und wie muß das Programm geändert werden, um das Feld voll auszunutzen?
- Durch eine Änderung in Zeile 11Ø kann das Feld "absteigend" sortiert werden.
- Wie muß das Programm geändert werden, um zuerst ein Feldelement mit allen anderen zu vergleichen?

## Reaktionstest

**Inhalt:** Der auf dem Bildschirm angezeigte Buchstabe ist über Tastatur einzugeben und danach ist die ENTER-Taste zu drücken. Es wird mit der Uhr des Rechners gemessen, wieviel Zeit benötigt wird, um 12 Buchstaben zu erkennen.

**Programm:**

```
10 WINDOW:CLS:WINDOW 23,23,0,39
20 POKE 30,0: POKE 31,0
30 FOR I=1 TO 12
40 B$ = CHR$(65 + INT(26 * RND(1)))
50 PRINT AT(12,19); B$
60 INPUT "Buchstaben eingeben!"; R$
70 IF B$ <> R$ THEN GOTO 60
80 NEXT I
90 S = 60 * PEEK(30) + PEEK(31)
100 WINDOW:CLS
110 PRINT "Gesamtzeit: "; S; "Sek."
120 PRINT "Zeit/Buchstabe: "; S/12; "Sek."
```

**Hinweis:** In Zeile 20 werden die Speicherzellen 30 und 31 zum Messen der Minuten bzw. Sekunden auf Null gesetzt. Die angezeigten Buchstaben werden in Zeile 40 ausgelost. RND(1) liefert einen zufälligen Wert zwischen 0 und 1. INT (argument) hat als Funktionswert die größte ganze Zahl, die im argument enthalten ist. Die ASCII-Codes für Buchstaben liegen zwischen 65 und 90. In Zeile 90 wird die Rechneruhr abgefragt.

**Variante:** Das Drücken der ENTER-Taste nach einer Eingabe kann unterbleiben, wenn Zeile 60 durch  
60 R\$ = INKEY\$  
ersetzt wird.

## Bilder malen

- Inhalt:** Durch entsprechende Positionierung von Grafikzeichen auf dem Bildschirm können Bilder und Figuren dargestellt werden.  
Der Cursor wird unter Nutzung der Kursortasten positioniert.
- Programm:** 1Ø WINDOW: CLS  
2Ø PRINT INKEY\$;: GOTO 2Ø
- Hinweise:** - Das Semikolon hinter INKEY\$ ist sehr wichtig!  
- Das Programm wird durch **RUN** gestartet und durch **STOP** beendet.
- Varianten:** Durch Nutzung der **COLOR** -Taste können auch mehrfarbige Bilder erstellt werden.  
Die Tastenkombination **CONTR F** läßt alle anschließend angezeigten Zeichen blinken.

## Zufällige Bilder

**Inhalt:** Durch Verwendung der Zufallszahlenfunktion RND werden zufällige Bilder erzeugt, die aus Grafikzeichen bestehen.

**Programm:**

```
1Ø WINDOW: CLS
2Ø FOR I = 1 TO 1000
3Ø N = INT(128 * RND (1)) + 128 :| Zeichencode
4Ø Z = INT(15 * RND (1)) + 5      :| Zeile
5Ø S = INT(30 * RND (1)) + 5     :| Spalte
6Ø PRINT AT(Z,S);CHR$(N)
7Ø NEXT I
```

**Hinweis:** Die Funktion RND (1) liefert Zufallszahlen im Intervall (0,1). Durch INT (X) wird der ganze Teil einer Zahl X berechnet, d.h. die größte ganze Zahl, die kleiner als X ist.

- Varianten:**
- Durch Änderung der Zeile 30 kann der benutzte Zeichensatz verändert werden, z.B. können nur die Grafikzeichen mit den Code-Nummern 240 ... 255 benutzt werden.
  - Man überlege, wie ein zufälliges Farbmuster erzeugt werden kann.  
(Farbcodes 1 ... 8, Farbanweisungen INK oder PAPER nutzen)

## Raupe

**Inhalt:** Demonstration der Wiederholungsanweisung (FOR... NEXT) mit variablen Anfangs- und Endwerten sowie Schrittweiten am Beispiel einer sich hin- und her-bewegenden "Raupe".

**Programm:**

```
10 WINDOW: CLS
20 START = 1: ZIEL = 38: SCHRITT = 1
30 FOR I = START TO ZIEL STEP SCHRITT
40 PRINT AT(11,I-SCHRITT); " "
50 PRINT AT(11,I+SCHRITT); CHR$(205)
60 PAUSE 2
70 NEXT I
80 BEEP: SCHRITT = -SCHRITT
90 I = START: START = ZIEL: ZIEL = I
100 GOTO 30
```

**Hinweis:** Abbruch des Programmes mit der STOP-Taste. Das in Zeile 50 auszugebende Grafikzeichen kann auch über die Tastatur eingegeben werden.

**Varianten:** 1. Geschwindigkeitsänderung der Raupenbewegung. Durch die Änderung

```
60 PAUSE 1
```

z.B. nur noch eine Zehntelsekunde Pause zwischen jedem Schritt.

2. Verlängerung des Raupenkörpers durch folgende Änderungen:

```
20 START = 3: ZIEL = 36: SCHRITT = 2
40 PRINT AT(11,I-SCHRITT); " "
50 PRINT AT(11,I+SCHRITT); CHR$(205)
+CHR$(205)
```

## Laufanweisungen mit und ohne "Vortest"

**Inhalt:** Die Laufanweisung FOR I = AN TO EN STEP S wird in jedem Fall einmal durchlaufen. Dies kann unerwünscht sein, wenn

oder  $AN > EN$  und  $S > 0$   
 $AN < EN$  und  $S < 0$

gilt. In diesen Fällen ist ein "Vortest" sinnvoll.

**Programm:**

```
10 INPUT "Eingabe AN, EN, S:"; AN, EN, S
20 V% = "N"; INPUT "VORTEST - J/(N)"; V%
30 IF V% = "N" THEN PRINT " ": GOTO 50
40 IF (EN-AN)*S < 0 THEN 90
50 FOR I = AN TO EN STEP S: PRINT I: NEXT I
80 PRINT
90 PRINT "ENDE DER LAUFANWEISUNG"
100 . GOTO 10
```

**Hinweis:** Das Programm wird durch STOP beendet.

**Varianten:** Weitere Probleme werden deutlich, wenn man mit  $AN = -0.9$ ,  $EN = 0.6$ ,  $S = 0.1$  rechnet. Wie kann man diesen (natürlichen) Rundungsfehlern eines Computers begegnen?



## Potenzieren

**Inhalt:** Das Programm realisiert die Berechnung  $y^x$  für natürliches  $x$ . Das Besondere daran ist, daß ein Algorithmus eingesetzt wird, der nicht die ungenaue Potenzfunktion verwendet, sondern der auf der  $x$ -maligen Multiplikation der Basis  $y$  mit sich selbst beruht. Die Anzahl der dazu notwendigen Multiplikationen wird minimiert.

**Programm:**

```
1Ø INPUT "REELLE ZAHL Y="; Y
2Ø INPUT "NAT. ZAHL X="; X
3Ø E = 1
4Ø IF X/2 = INT (X/2) THEN X = X/2:
   Y = Y*Y; ELSE X = X-1: E = Y*E
5Ø IF X > 0 THEN 4Ø
6Ø PRINT "Y HOCH X ="; E
7Ø END
```

**Varianten:**

- Für  $y = 0$  und  $x = 0$  liefert das Programm  $y^x = 0$ , was wegen des Verbotes von  $0^0$  falsch ist. Wie kann man dies verhindern?
- Man kann die Anzahl der durchlaufenden Zyklen ermitteln, indem folgende Ergänzungen eingegeben werden:

```
24 ZY = 0
44 ZY = ZY + 1
64 PRINT "ZYKLEN:"; ZY
```

**Beispiele:**

|              |            |                       |
|--------------|------------|-----------------------|
| $y = 12.234$ | $x = 23$ , | $y^x = 1.Ø3294E+25$ , |
| Zyklen: 8    |            |                       |
| $y = 23$     | $x = 16$ , | $y^x = 6.13261E+21$ , |
| Zyklen: 5    |            |                       |

## Melodienausgabe

### Inhalt:

Die Melodienausgabe erfolgt mit Hilfe des Kassetteninterface (Kanal 0 des CTC). Dabei wird der Kanal 0 als Zeitgeber entsprechend den DATA-Anweisungen der Zeilen 20 bis 70 programmiert, in Zeile 150 gestartet und in Zeile 170 gestoppt. Die Tonhöhe ergibt sich aus der Zeitkonstanten, die Länge aus der Verzögerung entsprechend Zeile 160.

```
10 |-----
20 DATA 101,1,107,1,120,1,107,1,101,2
30 DATA 135,2,101,1,107,1,120,1,107,1
40 DATA 101,2,135,2,90,1,101,1,107,1
50 DATA 120,1,135,2,135,1,120,1,120,1
60 DATA 107,1,107,1,101,1,101,1,90,1
70 DATA 90,1,80,1,80,1,90,1,90,1,101,2
80 |-----
90 FOR I=0 TO 30
100 READ H,L:L=L*100
110 GOSUB 150
120 NEXT
130 END
140 |-----
150 OUT 128,7: OUT 128,H
160 FOR Q=1 TO L: NEXT Q
170 OUT 128,3
180 RETURN
```

### Varianten:

- Andere Werte für Tonhöhe und Verzögerung in den DATA-Anweisungen und eine entsprechend veränderte Zeile 90 ergeben andere Melodien.
- Mit den Anweisungen  
85 OUT 136,128  
125 OUT 136,0  
werden die Töne vom eingebauten Summer erzeugt.
- Wie ist das Programm zu verändern, damit Pausen entsprechender Länge in die Melodie eingebaut werden können?

## 6.BASIC - Anweisung für Plotter - Funktion

|            |                |   |
|------------|----------------|---|
| MOVE       | x,y            | Stift zum Punkt (x,y) bewegen                                       |
| PLOT       | x,y            | Linie zum Punkt (x,y) zeichnen                                      |
| POINT      | x,y            | Punkt (x,y) zeichnen  |
| -----      |                |   |
| MOVE STEP  | x,y            | Stift zum Punkt (Ausgangspunkt + Koordinatendifferenz x,y) bewegen  |
| PLOT STEP  | x,y            | Linie zum Punkt (Ausgangspunkt + Koordinatendifferenz x,y) zeichnen |
| POINT STEP | x,y            | Punkt (Ausgangspunkt + Koordinatendifferenz x,y) zeichnen           |
| -----      |                |   |
| ZERO       | x,y            | Nullpunkt für Absolutkoordinaten auf Plotterkoordinate (x,y) legen  |
| -----      |                |   |
| HOME       |                | Stift zum Punkt (0,0) bewegen (entspricht MOVE 0,0)                 |
| -----      |                |   |
| LABEL      | A\$            | Zeichenkette A\$ von aktueller Stiftposition aus schreiben          |
| -----      |                |   |
| SIZE       | xx; xy, yx, yy | Parameter für Textausgabe festlegen (Parameter s. Zeichnung)        |
| -----      |                |   |

### Beispiel:

```

10 ZERO 0,0: HOME
20 PLOT 2500,0
30 PLOT 2500,1750
40 PLOT 0,1750
50 PLOT 0,0
60 MOVE 665,657
70 SIZE 45,0,0,96
80 LABEL "robotron KC 85/1"
90 HOME_
    
```

0, 1750

2500, 1750

SIZE 30, 0, 20, 48

SIZE -30, 0, 0, -48

SIZE 45, 0, 0, 48

SIZE 0, -30, 48, 0

SIZE 30, 0, 0, 48

SIZE 30, 0, 0, 96

SIZE 0, 30, -48, 0

SIZE 21, 21, -34, 34

SIZE -21, 21, -34, -34

SIZE -21, -21, 34, -34

SIZE 21, -21, 34, 34

SIZE 30, 0, -20, 48



0, 0

2500, 0

# KLEINCOMPUTER robotron KC 85/1

## 7. Programmierung in Assembler

Programmierung von Tastatureingaben und Bildschirmausgaben mit dem Kleincomputer robotron KC 85/1

Für diese Funktionen stellt das Betriebssystem sogenannte Systemrufe zur Verfügung; die im Anwenderprogramm aufgerufen werden können. Alle Systemrufe beginnen auf der Adresse 5 und sind durch ihre Rufnummer im C-Register zu unterscheiden.

Je nach Art des Systemrufes werden Parameter im E- bzw. DE-Register übergeben oder im A- bzw. BC-Register zurückgegeben. Wird bei der Abarbeitung ein Fehler festgestellt, so wird das CY-Flag gesetzt.

### Ruf 11: Tastaturstatus

Gibt im A-Register den Code der zuletzt gedrückten Taste zurück.

Beachte: - A = 0 bedeutet: Es wurde keine Taste gedrückt.  
- Es wird nicht auf einen Tastendruck gewartet.  
- Es erfolgt keine Anzeige des Tastendrucks.  
- Der Tastendruck bleibt im Computer gespeichert.

### Ruf 1: Tastatureingabe

Wartet auf einen Tastendruck und übergibt dann den Tastencode im A-Register.

Beachte: - Es erfolgt keine Anzeige d. Tastendrucks.  
- Der Tastendruck wird gelöscht.

### Ruf 10: Zeichenketteneingabe von Tastatur

Im DE-Register ist die Adresse eines Pufferspeichers zu übergeben, wobei im 1. Byte des Puffers die max. Länge der Zeichenkette stehen muß.

Im 2. Byte des Puffers wird die tatsächliche Länge d. eingegebenen Zeichenkette und ab 3. Byte die Zeichenkette selbst zurückgegeben.

Beachte: - die Tastendrücke werden angezeigt  
- die Zeichenkette ist mit ENTER abzuschließen

### Ruf 2: Bildschirmausgabe

Der im E-Register stehende Zeichencode wird an der aktuellen Kursorposition angezeigt.

### Ruf 9: Zeichenkettenausgabe an Bildschirm

Im DE-Register ist die Adresse der Zeichenkette zu übergeben, da ab aktueller Kursorposition angezeigt werden soll.

Beachte: - Die Zeichenkette ist mit einem Byte 00H abzuschließen.

# KLEINCOMPUTER robotron KC 85/1

## Kurzbedienungsanleitung IDAS

1. Aufruf: **I D A S** **ENTER**
2. Alten Text löschen: IDAS **N**
3. Programmeingabe: + **R E A D** **ENTER**  
DEVICE (T/CC)? **ENTER**

Eingabe d. Programmzeilen (ohne Zeilennummer!)

Die letzte Zeile ist mit **ENTER** **STOP** abzuschließen

4. Programm zur Kontrolle ansehen: **L I S T** **ENTER**
5. Bei eventuellen Korrekturen und Fehlermeldungen ist der Standbetreuer um Rat zu fragen.
6. Programmstart: wie bei den einzelnen Beispielen angegeben
7. Verlassen d. IDAS: + **C M** **ENTER**

# KLEINCOMPUTER robotron KC 85/1

## Beispiel 1: Zeichenkettenausgabe an Bildschirm

---

```
10 ZKAN: LD DE,ZK ;Adr.Zeichenkette in DE
20 LD C;9
30 CALL 5
40 RET
50 ZK: DB OCH ;Bildschirm löschen
60 DB 7 ;Ausgabe BEEP-Ton
70 DB 'Demonstration der ' ;sichtbare
80 DB 'Zeichenkettenausgabe';Zeichen
90 DB ODH
100 DB OAH ;Kursor auf neue Zeile
110 DB 0 ;Endekennzeichen für ZK
```

Start des Programms: **C A L L Z K A N** **ENTER**

---

Beachte die Wirkung der Steuerzeichen in den Zeilen  
50, 60, 90 und 100!

## Beispiel 3: Anzeige des A-Registerinhaltes als Hexadezimalzahl

---

```

10  AOUT:  LD      D;A           ;A merken
20          RRCA
30          RRCA
40          RRCA
50          RRCA
60          CALL   AOUT1        ;1.Ziffer: Bit 7...4
70          LD      A,D         ;A zurück
80          CALL   AOUT1        ;2.Ziffer: Bit 3...0
90          LD      E,'H'
100         JR      AOUT3-#
110  AOUT1: AND     OFH         ;Bit 3...0=Hex-Ziffer
120         CMP    10          ;Wandlung
130         JRC   AOUT2-#
140         ADD    7
150  AOUT2: ADD    '0'         ;Ziffer-- Zeichen
160         LD      E,A
170  AOUT3: LD      C;2        ;Zeichen anzeigen
180         CALL   5
190         RET
    
```

### Bedienung:

---

1. A-Register laden (z.B.: LD A 34H ENTER )
2. Programm starten mit: CALL AOUT ENTER

Beachte: Das A-Register kann nur mit Werten zwischen 0H und 0FFH geladen werden.



# KLEINCOMPUTER robotron KC 85/1

## Beispiel 2: Abbruch eine Programms mit der Stoptaste

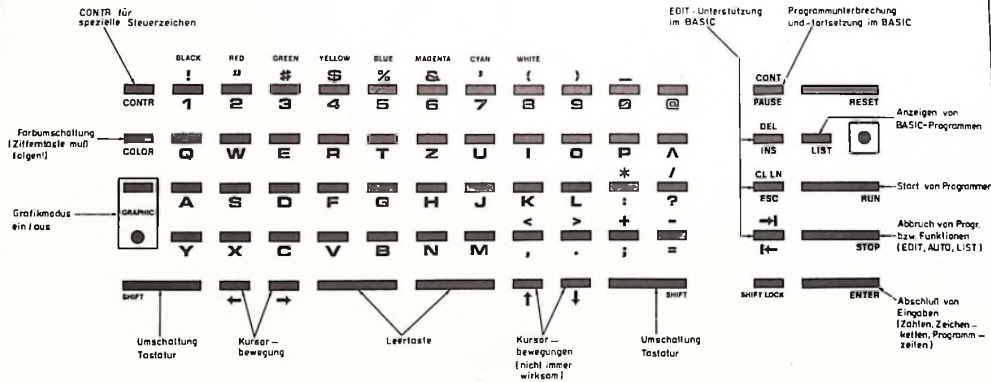
```

10  STOP: LD    E, '*'          ;Zeichen in E
20      LD    C, 2             ;Zeichenausgabe
30      CALL  5                ;an Bildschirm
40      LD    C, 11           ;Tastaturstatus abfragen
50      CALL  5
60      CMP   3                ;STOP gedrückt?
70      JRZ   STOP2-#         ;Ja-- STOP2
80      LD    HL, 1000H
90  STOP1: DEC  HL             ;HL=HL-1
100     LD    A, H
110     OR    L
120     JRNZ  STOP1-#         ;Schleife bis HL=0
130     JR    STOP-#          ;von vorn
140  STOP2: LD    C, 1         ;Tastendruck abholen
150     CALL  5
160     RET                    ;Ende

```

Start des Programms: **C A L L S T O P** **ENTER**

# Die Tastatur im Überblick



## Hinweise:

- Zur einfachen Eingabe von Zeichen (Buchstaben) müssen die Tasten kurz und kräftig gedrückt werden
- ständiges Drücken einer Taste führt zur Mehrfachanzeige des entsprechenden Zeichens bzw. zur wiederholten Ausführung der zugehörigen Funktion (z.B. INS, Kursorbewegungen).