



Teil 1

basic

für

Fortgeschrittene

Autor: Prof. Dr. H. Völz



Programme der Sendungen 1 - 5

<u>Name</u>	<u>Inhalt</u>	<u>Zähler</u>
FILE	erzeugt Zufallszahlen,-texte	
ZABUB	Zahlenbubble	
WOBUB	Wortbubble	
BUB2	Bubble mit Pointern, verkürzt	
SHAKER	Shakersort (hin und her)	
QUICK	Quicksort (binär)	
RUND	Runden von Zahlen	
REFOR	rechtsseitiges Formatieren	
PUFOR	Formatieren auf den Punkt	
FAKUL	Fakultät nach Stirling	
KOMBA	alle kombinatorische Formeln	
EULER	lange Eulersche Zahl	
PI	viele Ziffern von Pi	
TEFRA	testen für Fraktale	
FRAGEB	Fraktale als Gebirge + normal	
FRAKT	Fraktale in s/w hoher Auflösung	
FRAK87	Felder für Fraktale KC 87	
FRAK3	Fraktale: Felder von FRAK87	
FRAKTAL	farbige Fraktale	
RASTER	Raster-Hintergrund mit Farbbyte	
FNSA	bis 26 Fkt. direkt eingeben	
FUNKTION	direkte Fkt.-Eingabe + Tabelle	
PLOT3D	3D-Perspektive von Fkt.	
BASUHR	BASIC mit Uhr und Zeitmessung	
COPY	Bildschirmcopy !! LOAD !!	
BASUR	BASIC-Uhr für KC 87	
BINOM	Binomial - Koeffizient	
FIBON	Fibonacci - Zahlen	
GGT	größter gemeinsamer Teiler	
ACK1	Ackermann - Funktion (rekursiv)	
ACK2	Ackermann - Funktion (mit Stack)	

ZUSAMMENFASSUNGEN

=====

Sendung 1: Sortieren

1. Dateien oder Files sind eine Zusammenfassung aus mehreren Teilen, dies sind Abschnitte, Sätze, Worte oder Zahlen.
2. Sie lassen sich nach mehreren Gesichtspunkten sortieren. Zwei davon sind:
 - * Nach der Größe vorhandener Zahlen oder
 - * nach dem Alphabet.
3. In BASIC ist eine alphabet-ähnliche Sortierung leicht zu realisieren. Dabei wird die ASCII-Nummer als Kennzeichen benutzt. Dadurch stehen alle Kleinbuchstaben gemeinsam hinter den Großbuchstaben.
4. Die Eigenschaften der Sortiervverfahren betreffen vorrangig drei Punkte:
 - * Die Sortierzeit,
 - * Den Organisationsaufwand, d.h. die Größe des Programms,
 - * Den zusätzlich benötigten Speicherplatz, z.B. für die Anzahl der gebildeten Stapel.
5. Vor dem Sortieren kann die Datei unterschiedliche Zustände besitzen. Sie haben u.a. Einfluß auf die Sortierzeit. Sie kann:
 - * bereits richtig sortiert sein,
 - * genau entgegengesetzt sortiert sein,
 - * zufällig angeordnet sein.
6. Das einfachste Sortiervverfahren ist Bubble-Sort. Sein Algorithmus lautet:
 - A: Es wird mit dem ersten Wort, also $n=1$ der Datei begonnen.
 - B: Es wird immer das n -te Wort mit dem $(n+1)$ -ersten verglichen.
 - C: Sind beide richtig angeordnet, so geht es bei E weiter.
 - D: Sie sind falsch angeordnet, so werden die Worte vertauscht und es wird ein Flag gesetzt.
 - E: Es wird n um 1 erhöht. Ist das letzte Wort erreicht, so wird bei F, sonst bei B fortgefahren
 - F: Wenn das Flag gesetzt ist, beginnt man wieder bei A, sonst ist die Datei sortiert.
7. Es gibt zwei typische Operationen beim Sortieren:
 - * Den Vergleich und
 - * Den Austausch.Beide benötigen Zeit. Der Austausch kann bei Worten (im Gegensatz zu Zahlen) sehr nachteilig durch die Garbage-Collection beeinflußt werden. Dies kann vermieden werden, wenn der Austausch nur symbolisch erfolgt. Zu diesem Zweck nutzt man Zeiger engl. Pointer, die auf die richtige Reihenfolge der Worte verweisen. Nur sie werden dann beim Sortieren verändert.
8. Beim Bubblesort gibt es mehrere Varianten, die den Algorithmus zwar komplizierter, dafür aber schneller machen. Hierzu gehören u.a.:
 - * In der Datei wird nur bis zur erforderlichen Länge sortiert.
 - * In der Datei wird abwechselnd vorwärts und rückwärts sortiert.
- Im zweiten Fall wird von SHAKER-Sort gesprochen
9. Alle Sortiervverfahren unterscheiden sich u.a. dadurch, über welchen Abstand hinweg Worte verglichen und/oder ausgetauscht werden.
10. Ein sehr schnelles, aber kompliziertes Verfahren ist QUICK-Sort. Sein Algorithmus läuft wie folgt ab:

- A: Das ganze File wird zunächst als ein Stapel betrachtet.
 - B: Aus den Worten eines Stapels wird in etwa ein mittlers als Bezug ausgewählt.
 - C: Es werden zwei Stapel gebildet, welche jeweils Worte enthalten, die links bzw. rechts vom Bezug stehen.
 - D: Für jeden Stapel wird iterativ der Vorgang von B beginnend solange wiederholt, bis nur ein oder zwei Worte existieren.
 - E: Nun werden die Worte schrittweise von den Stapeln richtig hintereinander angeordnet.
- Es sei bemerkt, daß im allgemeinen schrittweise immer nur ein Stapel bis auf ein oder zwei Worte geteilt wird. Der Stand der anderen Stapel wird zwischendurch auf einem Stack gespeichert.

Sendung 2: Zahlen

1. Runden bedeutet immer, Zahlen auf eine bestimmte Genauigkeit festzulegen. Dazu diene ein Stellenwert X, z.B. 100, 10, 0.1 oder 1/100. Dann wird der folgende Algorithmus genutzt:

A: Der vorhandene Zahlenwert wird durch X dividiert.

B: Es wird 0.5 dazu addiert.

C: Es wird der Integerwert gebildet.

D: Der neue Zahlenwert wird mit X multipliziert.

2. Formatieren liegt immer dann vor, wenn Zahlen, Worte oder Texte in ein Schema, eine Tabelle oder ein Formular eingepaßt werden müssen. Bei Zahlen bedeutet dies außerdem, daß sie auf genau definierte Positionen zu setzen sind.

3. Wenn verschiedene Elemente, z.B. A, B, C und D einzeln oder mehrfach existieren, so können sie in unterschiedlicher Weise zusammengefaßt werden. Es können aus ihnen also Worte verschiedener Länge und Gestalt erzeugt werden. Entsprechend den dabei verwendeten Regeln sind zu unterscheiden:

* Permutationen, Variationen, Kombinationen

In den drei Fällen sind außerdem zu unterscheiden, ob sich gleiche Elemente in einem Wort wiederholen dürfen oder nicht.

Bei den PERMUTATIONEN müssen alle vorhandenen Elemente zum Aufbau eines Wortes benutzt werden. Wenn gleiche Elemente existieren, sind AABCD, ABACD, DABBC usw. gültige Worte. Anderfalls nur ABCD, BCAD, CDAB usw. In diesem Fall beträgt die Anzahl der Permutationen $n!$, also hier $4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$.

Bei den VARIATIONEN werden aus n Elementen Gruppen mit k Elementen bei k'n gebildet. Hier ist die Reihenfolge der Elemente wichtig. Sollen aus den 4 Elementen A, B, C, D drei ausgewählt werden, so sind im ersten Fall AAA, AAB, ACA unterschiedliche Worte. Dann gibt es n^k mögliche Variationen. Andernfalls werden Elemente der Art AAB, ABA und BAA als gleich betrachtet. Sie werden dann als nur ein Element gezählt.

Bei den KOMBINATIONEN werden im Gegensatz zu den Variationen nur unterschiedliche Elemente zugelassen. Ohne Unterscheidung der Reihenfolge sind also ABC, BCA, BAC verschiedene Worte. Ansonsten gelten sie als gleiche Worte. Dann existieren n über k Kombinationen.

4. Es können folgende Zahlen unterschieden werden:

* Ganze Zahlen, auch Integer genannt, wie 3, 127 oder -68.

* Brüche, rationale Zahlen, wie $1/3$, $1/385$ usw. Ihre Dezimaldarstellung bricht irgendwann ab oder führt zu Perioden, wie z.B. $1/3 = .3333333... \text{ oder } \frac{1}{3}$

1/7 = 0.142857 142857 142857

- * Irrationale Zahlen. Sie sind durch einen unendlich langen Dezimalbruch ohne Periode gekennzeichnet. Beispiele hierfür sind $\sqrt{2}$, π oder die Eulersche Zahl e .

Sendung 3: Fraktale

1. Die Anweisung $A = X * A * (A - 1)$ besitzt bei einem Startwert $A = .5$ in einer Iterationsschleife je nach X das folgende Verhalten:
 - # Für $0 < X \leq 1$ konvergiert A gegen 0, wobei die Geschwindigkeit hierfür beachtlich von X abhängt.
 - # Für $1 < X < 1.7$ zeigt A ein periodisches Verhalten. Nach einem Einschwingvorgang werden 2, 4, 8 oder auch mehr Zahlenwerte periodisch angenommen.
 - # Für $1.7 < X < 2$ verhält sich A stochastisch, d.h. es ist keine Regelmäßigkeit in der Abfolge der A zu erkennen.
 - # Für $X > 2$ wächst A über alle Grenzen. Es divergiert gegen unendlich. Beim Rechner tritt folglich Überlauf auf. Die Geschwindigkeit der Divergenz ist wieder von X abhängig.
2. Den Fraktalen liegt folgendes Prinzip zu Grunde:
 - A: Es wird ein X - und Y -Bereich für die Eingangsgrößen gewählt. Dieser wird punktuell auf die Pixel des Bildschirms abgebildet.
 - B: Es werden zwei Formeln genutzt:
 $A = F(A, B, X, Y)$ und $B = G(A, B, X, Y)$
 - C: Es werden die zwei Iterationsvariablen A und B mit Startwerten belegt.
 - D: Bei der Iteration werden die Konvergenz- und/oder Divergenzgeschwindigkeit bestimmt.
 - E: Aus diesen Geschwindigkeiten wird die Färbung der Pixel festgelegt.

Für ein stabiles Verhalten ist es günstig, eine maximale Zykluszahl und eine größte Zahl für die Divergenz festzulegen.

3. Die Berechnungen bei den Fraktalen dauern für jeden Punkt und damit erst recht für die Fläche sehr lange. Weiter ist bei neuen Bereichen und Formeln vor der Rechnung nichts über das Aussehen des Fraktals bekannt. Deshalb sind insbesondere bei Kleinrechnern Testmethoden erforderlich. Hierfür wurde das Programm TEFRA entwickelt.

4. Anschaulich ist die Konvergenzgeschwindigkeit auch als Höhe eines Gebirges zu interpretieren. Je mehr Zyklen benötigt werden, desto höher ist hier der Berg. Eine Darstellung mit verdickten Linien ermöglicht das Programm FRAGEB. Es enthält eine zweite Routine, die dasselbe Gebiet als Fraktal darstellt. Der Vergleich beider Bilder ermöglicht ein tieferes Verständnis der Problematik.

Sendung 4: Maschinenprogramme in BASIC

1. Maschinenprogramme können in DATA-Zeilen eines BASIC-Programmes abgelegt werden. Es ist vorteilhaft, wenn der übliche HEX-Code statt Dezimalzahlen verwendet wird. Dazu ist vor dem Poken an die freien Speicherstellen der HEX-Wert in eine Dezimalzahl zu wandeln. Günstig ist es, den HEX-Code halbbyteweise in zwei ASCII-Zeichen zu zerlegen. Über den ASCII-Code erfolgt die Berechnung der Dezimalzahl.
2. Eine fast ideale Methode beruht darauf, den Maschinencode in

einer REM-Zeile abzulegen. Hierbei sind folgende Fakten zu beachten

- # Um Adreßverschiebungen beim Programmieren in BASIC zu vermeiden, ist die erste Zeile, also Zeile 0 besonders geeignet.
- # Das Maschinenprogramm darf keine Nullen enthalten.
- # Statt des REM zu Beginn kann auch "!" oder GOTO n stehen.
- # Extrem lange Maschinenprogramme können in einer überlangen Zeile abgelegt werden. Dann muß manuell der erste Zeilenpointer richtig gesetzt werden.
- # In der Zeile mit dem Maschinencode lassen sich auch gut Copyright und andere Kommentare unterbringen.

Mit dieser Methode lassen sich Programme erzeugen, die gleichermaßen auf allen KC-Rechnern laufen.

3. Für Maschinenprogramme nutzbarer Speicherraum kann auch an folgenden Stellen gewonnen werden:

- # Der Anfang von BASIC-Programmen ist verschiebbar. Dies erfolgt durch verändern eines Pointers.
- # Am Speicherende kann Freiraum mittels Pointer oder der Anweisung CLEAR geschaffen werden.
- # Hinter dem Speicherraum für die Felder ist meist ein Bereich bis zum BASIC-Stack nutzbar.

Hier eine Zusammenfassung der nutzbaren Speicherbereiche, wobei das BASIC-Programm normal von 400H ab existiert:

=====			
! Prinzipiell freier Speicherraum !			

! KC 85/2 und 3 !		! KC 85/1 und KC 87 !	

! 0 ... 140H		! 20C...2BF sehr bedingt !	
! 200...2FFH		! 2C0...2FF falls ohne Erw.!	
! 400...3FFFH/7FFFH 1)		! 400...3FFFH/7FFFH/8FFFH 1)	
! B800...BBFFH sehr bedingt 2)		! EBC0...EBFFH wenn Farb- !	
! BC00...BFFFH 2)		! modul !	

1) nutzbar über:

- a) REM-Zeilen.
- b) CLEAR XX,YYYY.
- c) Programmverschiebung 35F/60.
- d) bedingt hinter dem Array-Bereich.

2) mit beachtlich verlängerter Zugriffszeit.

Sehr bedingt bedeutet, daß hier eventuell aus dem System heraus Probleme auftreten können.

Sendung 5: Rekursiv

1. Für viele Funktionen ist es vorteilhaft, sie rekursiv zu definieren. Dabei steht sowohl rechts als auch links vom Gleichheitszeichen die zu definierende Funktion. Sie zitiert sich also selbst. Auf diese Weise kann Wesentliches über die Funktion besonders deutlich hervorgehoben werden. Beispiele sind die Fakultät, der Binomialkoeffizient und die Fibonacci-Zahlen.

2. Die Programmierung von Funktionen, die rekursiv definiert sind ist meist mittels FOR-NEXT-Schleifen, also iterativ möglich.

3. Bei der iterativen Berechnung von rekursiv definierten Funktionen können zwei Fälle unterschieden werden.:

- # Die Anzahl der Iterationszyklen ist zu Beginn bekannt.

Dann genügt die einfache FOR-NEXT-Schleife.

- # Die Anzahl der Iterationszyklen ist - wie beim euklidischen Algorithmus für den größten gemeinsamen Teiler - zu Beginn nicht bekannt. Dann existiert aber eine Ende-Bedingung. Dies entspricht der WHILE-WEND-Konstruktion anderer Sprachen. Sie wird in BASIC durch die FOR-NEXT-Schleife mit der Schrittweite Null realisiert.

4. Mittels Rekursion und ergänzender Aussagen lassen sich nicht nur einzelne Funktionen, sondern ganze Klassen von Funktionen definieren. Dadurch ist es möglich, alles was berechenbar ist, zu erfassen.

5. Eine Funktion, die in diesem Kontext besondere Bedeutung hat, ist die Ackermann-Funktion:

$A(0,y)=y+1$

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

$A(x,y)=A(x-1,A(x,y-1))$

Sie gehört zur Klasse der allgemein rekursiven Funktionen. Zu ihrer Berechnung ist daher eigentlich eine rekursive Programmiersprache notwendig. In BASIC ist dies mittels eines nachgebildeten Stacks möglich. Dies leistet das Programm ACK2.

6. Durch einem Trick ist ein Weg mittels Arrays zu realisieren. Er läßt sogar die Struktur der Ackermann-Funktion deutlich hervortreten. Darüber hinaus ist das Programm ACK1 sogar extrem viel schneller als das eigentlich rekursive Programm. Es verlangt erheblich weniger organisatorischen Aufwand.

7. Rekursives Programmieren nutzt die Vorteile der rekursiven Definitionen, erfordert aber meist lange Abarbeitungszeiten, die infolge der umfangreichen Stackarbeit entstehen. Deshalb sind iterative Verfahren oft vorteilhaft. Sie sind jedoch nicht für alle rekursiven Probleme nutzbar. Dies gilt insbesondere für rekursive Prozeduren und Strukturen.

```

2 WINDOW:CLS:S=32:GOSUB58:CLEAR64:IFFRE(X)<4000THEN110
4 CLEAR3750:DIMT$(30):PRINT:PRINT:INPUT"Bildschirm Zeilen,Spalten: ";Z,S
6 GOSUB58:WINDOW4,14,9,18:PRINT"1=EINGEBEN2=ANZEIGEN3=EINLESEN4=AUSGEBEN";
8 IFFRE(X)>475THENPRINT"5=AENDERN 6=DRUCKEN ";
10 PRINT"0=ENDE":PRINT:INPUT"AUSWAHL: ";A
12 IFA=0THENWINDOW0,Z-1,0,S-1:CLS:WINDOW:CLS:END
14 WINDOW2,Z-1,0,S-1:CLS:ONAGOTO16,28,52,54,60,88
16 PRINT"KEIN KOMMA!":I=0:J=0:PRINTAT(1,3);"Zeile:      Block:      Frei:"
18 T$(J)="" :K=0:PRINTAT(1,19);J+1
20 PRINTAT(1,9);I+1:S$="" :INPUT"";S$:K=K+LEN(S$):T$(J)=T$(J)+S$+" "
22 PRINTAT(1,28);FRE(X$)-300:IFK>120THENIFJ<40ANDS$<>"$"THENJ=J+1:GOTO18
24 IFS$="$$"ORJ>=30THENCLS:GOTO6
26 I=I+1:GOTO20
28 I=0:J=0:N=0
30 IFJ<=30THENS$=T$(J):K=0:ELSEINPUT"";A$:CLS:GOTO6
32 IFLEN(S$)=0THENJ=J+1:GOTO30:ELSEL=INSTR("$",S$):M=INSTR(" ",S$)
34 IFL>0ANDM>L THENNM=L
36 IFN+M>STHENGOSUB46:IFA$="$"THENCLS:GOTO6
38 PRINTLEFT$(S$,M-1);:IFM>M+NTHENPRINT" ";
40 N=N+M:IFLEN(S$)>MTHENS$=MID$(S$,M+1):ELSEJ=J+1:GOTO30
42 IFS$="$$"THENJ=41:GOTO30:ELSEIFM<>LANDN<STHEN32
44 GOSUB46:IFA$<>"$"THEN32:ELSECLS:GOTO6
46 A$="" :I=I+1:N=0:IFN<STHENPRINT
48 IFI/(Z-3)>INT(I/(Z-3))THENRETURN
50 A$=INKEY$:IFA$=""THEN50:ELSERETURN
52 GOSUB56:CLOAD*""+S$;T$:CLS:GOTO6
54 GOSUB56:CSAVE*""+S$;T$:CLS:GOTO6
56 INPUT"DATEINAME: ";S$:RETURN
58 PRINTAT(0,0);" MINTEX/3 (C) bidat'87":PRINTAT(1,0);STRING$(S,"="):RETU
60 INPUT"BLOCK: ";J:IFJ=0THEN80
62 J=J-1:PRINTAT(1,3);"Block:      Frei:"
64 IFJ<=30THENS$=T$(J):K=0:ELSECLS:GOTO6
66 PRINTAT(1,9);J+1:PRINTAT(1,18);255-LEN(T$(J))
68 IFLEN(S$)=0THENJ=J+1:GOTO64:ELSEM=INSTR(" ",S$)
72 S$=LEFT$(S$,M):A$=S$:PRINTA$;:INPUT"";A$:IFLEFT$(A$,1)="$"THENJ=41:GOTC
74 IFA$<>S$THENT$(J)=LEFT$(T$(J),K)+A$+MID$(T$(J),K+M)
76 K=K+LEN(A$):IFLEN(T$(J))>KTHENS$=MID$(T$(J),K+1):ELSEJ=J+1:GOTO64
78 IFS$<>"$$"THEN66:ELSECLS:GOTO6
80 PRINT"TEXT ANFUEGEN":J=0
82 K=INSTR("$",T$(J)):IFK=0ANDJ<30THENJ=J+1:GOTO82
84 IFJ>=30ORFRE(X$)<300THENCLS:PRINT"SPEICHER VOLL!":GOTO6
86 T$(J)=LEFT$(T$(J),K-1):GOTO20
88 INPUT"Druckseite Zeilen,Spalten: ";Y,X:I=0:J=0:N=0
90 IFJ<=30THENS$=T$(J):K=0:ELSEPRINT#2CHR$(12);:CLS:GOTO6
92 IFLEN(S$)=0THENJ=J+1:GOTO90:ELSEL=INSTR("$",S$):M=INSTR(" ",S$)
94 IFL>0ANDM>L THENNM=L
96 IFN+M>XTHENGOSUB106:IFA$="$"THENCLS:GOTO6
98 PRINT#2LEFT$(S$,M-1);:IFM>M+NTHENPRINT#2" ";
100 N=N+M:IFLEN(S$)>MTHENS$=MID$(S$,M+1):ELSEJ=J+1:GOTO90
102 IFS$="$$"THENJ=41:GOTO90:ELSEIFM<>LANDN<XTHEN92
104 GOSUB106:IFA$<>"$"THEN92:ELSECLS:GOTO6
106 A$="" :I=I+1:N=0:IFN<XTHENPRINT#2" "
108 IFI/Y>INT(I/Y)THENRETURN:ELSEPRINT#2CHR$(12);:INPUT"";A$:RETURN
110 PRINT"ZU WENIG SPEICHER! NEU STARTEN!":DELETE60,110

```

↳ 2

KURZBESCHREIBUNG MINTEX/3

=====

Rechnertyp: alle KC85 kompatiblen
Programmiersprache: HC-BASIC
Speicherbedarf: 6 KBytes

FUNKTION:

MINTEX/3 ist ein minimaler in BASIC geschriebener Texteditor der fuer die Bildschirmformate der KC85 kompatiblen Kleincomputer eingestellt werden kann. Das Programm ermoeeglicht bei einem verfuegbaren Anwenderspeicher von 5 KBytes die Texteingabe und Anzeige sowie das Abspeichern auf Kassette. Stehen mindestens 6 KBytes zur Verfuegung sind zusaetzlich Textaenderungen und Textausdruck moeglich. Der Textspeicher kann ca. 3.5 KBytes (entspr. 1 Druckseite) aufnehmen. In dem nur 2 KBytes grossen Programm koennt natuerlich keine besonders komfortablen und von anderen Textsystemen her bekannten Funktionen realisiert werden. Es soll den KC85 Nutzern die Erstellung und Weitergabe von Programmdokumentationen auf Kassette ermoeeglichen.

BEDIENUNG:

Das Programm wird mit CLOAD"MINTEX" geladen und mit RUN gestartet. Stehen weniger als 6 KBytes Anwenderspeicher zur Verfuegung erfolgt die Ausschrift ZU WENIG SPEICHER! NEU STARTEN! Dabei werden die Betriebsarten AENDERN und DRUCKEN zur Speicherplatzgewinnung geloescht. Es ist nochmals RUN einzugeben.

Es wird nun das Bildschirmformat abgefragt. Zeilen- und Spaltenzahl sind durch Komma getrennt und mit ENTER abgeschlossen einzugeben. Nun erscheint das Betriebsartenmenue aus dem die gewuenschte Betriebsart durch Eingabe ihrer Kennzahl und ENTER auszuwaehlen ist.

1=EINGABE

Zunaechst wird der Hinweis ausgegeben dass der Text KEIN KOMMA enthalten darf. Danach kann die Texteingabe beginnen. In einer Statuszeile wird die Nummer der Eingabezeile des Blockes und der freie Speicherplatz angezeigt. Die Blocknummer kann maximal 31 erreichen. Eine Eingabezeile kann unabhaengig vom Bildschirmformat 72 Zeichen enthalten. Darin sind Aenderungen durch Cursorverschieben ueberschreiben und Benutzung der Tasten INS und DEL moeglich. Mit ENTER wird die Zeile uebernomen. Eine besondere Bedeutung hat das Add-Zeichen (Affenschwanz Klammeraffe). Ein Affenschwanz erzwingt bei Anzeige und Ausdruck einen Zeilenwechsel und zwei Affenschwaenze kennzeichnen das Textende.

2=ANZEIGEN

Der im Speicher stehende Text wird dem Bildschirmformat angepasst angezeigt. Nach Fuellen des Bildschirms wird auf eine Tastenbetaetigung gewartet wobei \$ die Betriebsart beendet. Am Textende wird mit ENTER das Menue erreicht.

3=EINLESEN

Nach Eingabe des Dateinamens und Betaetigung der ENTER-Taste wird ein Text von Kassette eingelesen.

4=AUSGEBEN

Nach Eingabe des Dateinamens und Betaetigung der ENTER-Taste wird der im Speicher stehende Text auf Kassette ausgegeben.

5=AENDERN

Es ist die Blocknummer fuer den zu aendernden Textbereich einzugeben. Null ermoeeglicht ein Weiterschreiben am Textende. Ansonsten wird der Blockinhalt wortweise angezeigt. Mit ENTER wird das Wort unveraendert uebernommen mit \$ die Betriebsart abgebrochen. Ansonsten kann ein anderer Text von maximal der als frei angezeigten Laenge bzw. max. 72 Zeichen eingegeben werden. (VORSICHT Dabei kann der Gesamtspeicher ueberlaufen!)

6=DRUCKEN

Zunaechst ist das Druckformat in der Form Zeilenzahl Komma Spaltenzahl einzugeben. Danach erfolgt der Ausdruck gegebenenfalls mit Stop nach jeder Seite.

0=ENDE

Beendet die Abarbeitung des Programms MINTEX.

MINDOK von Biener

```

0 REM: Maschinencode
10 DEF FN A(X)=X+6*INT(X/10)
20 CLS: PRINT TAB(10)"## BASUHR ##": PRINT
30 INPUT"1: ein- 2: ausschalten 3: Stoppuhr";A
40 ON A GOSUB 60,90,110: GOTO 30
50 !----- einschalten -----
60 INPUT"h,m,s=";A, B, C: POKE 257, FN A(A): POKE 258, FN A(B)
70 POKE 259, FN A(C): CALL*406: RETURN
80 !----- ausschalten -----
90 CALL*497: RETURN
100 !----- Stoppuhr -----
110 INPUT"Zyklen=";Z: CALL*49E
120 FOR X=1 TO Z: NEXT: CALL*49E: RETURN
130 ! in 257 Stunden: 258 Minuten: 259 Sekunden: 260 1/100 Sekunden

```

```

10 CLS: PRINT TAB(13)"### BASUR ###": PRINT
20 INPUT"1: stellen 2: anzeigen 3: stoppen";A
30 ON A GOSUB 40, 50, 60: GOTO 20
40 INPUT"h,m,s=";A, B, C: POKE 29,A: POKE 30,B: POKE 31,C: RETURN
50 FOR I=1 TO 3: PRINT AT(0,25+3*I);PEEK(28+I): NEXT: GOTO 50
60 INPUT"Zyklen=";Z: CALL*F1B8
70 FOR X=1 TO Z: NEXT: CALL*F1B8: RETURN

```

COPY

```

10 CLEAR 30, 6140: WINDOW 0,31,0,39: CLS
20 FOR X=3 TO 350 STEP 6
30 CIRCLE X+5, X+80, X, 7
40 NEXT: CALL*400: INPUT";A
50 !alles fuer 16 K: COPY=CALL*432
60 !32 K: CLEAR 30, 22520: 406: 438
70 !SAVE in CAOS 300: (3D7/8): C08C
80 !Eigene Programme Start:35F/60

```

!!!!LOAD in CAOS!!!!

```

10 PRINT TAB(10);"## BUB2 ##": PRINT
20 CLEAR 10000: GOSUB 900: INPUT "Start";A
30 FOR X=0 TO N: F=0: FOR Y=0 TO N-1-X
40 IF A$(A(Y))>A$(A(Y+1)) THEN A=A(Y): A(Y)=A(Y+1): A(Y+1)=A: F=1
50 NEXT: PRINT X;: IF F=0 THEN X=N
60 NEXT
70 FOR X=0 TO N: PRINT A$(A(X)): NEXT: END
890 REM----- TEXTE -----
900 INPUT "Startwert > 0 ";A: A=RND(-A)
910 A=ASC("A"): B=ASC("Z")-A+1
920 INPUT "   Anzahl   ";N: DIM A$(N),A(N)
930 FOR X=0 TO N: A$=""
940 : FOR Y=0 TO 20*RND(1)
950 :   A$=A$+CHR$(A+B*RND(1))
960 NEXT: A$(X)=A$:A(X)=X: NEXT: RETURN

```

```

10 PRINT TAB(10);"## SHAKER ##": PRINT
20 CLEAR 10000: GOSUB 900: INPUT "Start";A
30 FOR X=0 TO N STEP 2: F=0: X2=INT(X/2): FOR Y=X2 TO N-1-X2
40 IF A$(A(Y))>A$(A(Y+1)) THEN A=A(Y): A(Y)=A(Y+1): A(Y+1)=A: F=1
50 NEXT: PRINT X;: IF X=0 THEN X=N: GOTO 90
60 : F=0: FOR Y=N-2-X2 TO X2 STEP -1
70 IF A$(A(Y))>A$(A(Y+1)) THEN A=A(Y): A(Y)=A(Y+1): A(Y+1)=A: F=1
80 NEXT: PRINT X+1;: IF F=0 THEN X=N
90 NEXT
100 FOR X=0 TO N: PRINT A$(A(X)): NEXT: END
890 REM----- TEXTE -----
900 INPUT "Startwert > 0 ";A: A=RND(-A)
910 A=ASC("A"): B=ASC("Z")-A+1
920 INPUT "   Anzahl   ";N: DIM A$(N),A(N)
930 FOR X=0 TO N: A$=""
940 : FOR Y=0 TO 20*RND(1)
950 :   A$=A$+CHR$(A+B*RND(1))
960 NEXT: A$(X)=A$:A(X)=X: NEXT: RETURN

```

```

10 PRINT TAB(15)"## RASTER ##": PRINT
20 RESTORE: FOR X=0 TO 70: READ A$
30 A=ASC(LEFT$(A$,1))-48: IF A>9 THEN A=A-7
40 B=ASC(RIGHT$(A$,1))-48: IF B>9 THEN B=B-7
50 POKE X,A*16+B: NEXT
60 INPUT"Raster";A: CALL*0: GOTO60
70 DATA F5,E5,D5,C5,DB,88,CB,D7,D3,88,21,00,A8,06,10,7E,EE,04,77,23,23
80 DATA 10,F8,11,40,00,19,E5,B7,11,00,80,ED,52,E1,38,E8,06,40,2B,10,FD
90 DATA 06,04,7E,EE,04,77,23,23,10,F8,11,10,00,19,E5,B7,11,00,B2,ED,52
100 DATA E1,38,E8,C1,D1,E1,F1,C9

```

FNSA

```

0 REM: Maschinencode
10 DEF FN A(X)=X*X:REMREM:.....
20 DEF FN B(X)=X^2:REMREM:.....
30 DEF FN C(X)=COS(X):REM:.....
40 DEF FN D(X)=TAN(X):REM:.....
50 DEF FN E(X)=ATN(X):REM:.....
60 B=1032: FOR A=65 TO 69: A$=""
70 PRINT"FN "CHR$(A)"(X)="";INPUT"";A$: IF A$>"" THEN POKE B,A: CALL*
80 NEXT

```

40A

```

0 REM: Maschinencode
10 DEFFNA(X)=LN(X)/LN(2):REM:.....
20 WINDOW 0,31,0,39: CLS: PRINT TAB(10)"### FUNKTION ###": PRINT
30 INPUT"F(X)="";A$: CALL*40A: PRINT
40 PRINT TAB(6);"Tabelle": PRINT: PRINT"X=",A$: PRINT
50 FOR X=1 TO 10: PRINT X, FN A(X): NEXT

```

PLOT3D

```

0 GOTO 560: MASCHINENCODE
10 !----- Geraden -----
20 LINE XS,YS,XZ,YZ,0: XS=XZ: YS=YZ: RETURN
30 !----- Schnittpunkte -----
40 H=(W0-T)/(V+W0-T-W1): S=S+H*(U-S): T=T+H*(V-T)
50 IF T>254 THEN T=254
60 IF T<1 THEN T=1
70 RETURN
80 !---- Ober,unter-Grenze ---
90 W1=W0: GOSUB 40: XS=S: YS=T: RETURN
100 W1=W0: GOSUB 40: XZ=S: YZ=T: GOTO 20
110 !----- algem. Grenzen ---
120 K=H: W1=G(B,A): W0=G(B,A-1)+H*(W1-G(B,A-1))
130 GOSUB 40: H=H+K-H*K: RETURN
140 !----- Horizonte -----
150 IF W>G(1,A-1) THEN G(1,A-1)=W+.5
160 IF W<G(0,A-1) THEN G(0,A-1)=W-.5
170 NEXT: NEXT: GOTO 690
180 REM----- RECHNUNG -----
190 COLOR1,7:::CLS:C=L:REM88AA
200 A1=-.5:REM:.....
210 FOR Y=YU TO YO STEP (YO-YU)/L

```

```

10 PRINT TAB(10);"## QUICK ##": PRINT
20 CLEAR 10000: DIM S(18,1): GOSUB 900: INPUT"Start";A
30 X=0: S(0,0)=0: S(0,1)=N
40 L=S(X,0): R=S(X,1): X=X-1
50 Y=L: J=R: H=A(INT((R+L)/2))
60 IF A$(A(Y))>=A$(H) GOTO 90
70 IF Y>R GOTO 90
80 Y=Y+1: GOTO 60
90 IF A$(A(J))=A$(H) THEN 120
100 IF J<=L GOTO 120
110 J=J-1: GOTO 90
120 IF Y<=J THEN M=A(Y): A(Y)=A(J): A(J)=M: Y=Y+1: J=J-1
130 IF Y<=J GOTO 60
140 IF (R-Y)<=(J-L) GOTO 170
150 IF L<J THEN X=X+1: S(X,0)=L: S(X,1)=J
160 L=Y: GOTO 190
170 IF Y<R THEN X=X+1: S(X,0)=Y: S(X,1)=R
180 R=J
190 IF R>L GOTO 50
200 IF X>=0 THEN PRINT X;: GOTO 40
210 FOR X=0 TO N: PRINT A$(A(X)): NEXT: END
890 REM----- TEXTE -----
900 INPUT"Startwert> 0 ";A: A=RND(-A)
910 A=ASC("A"): B=ASC("Z")-A+1
920 INPUT" Anzahl ";N: DIM A$(N), A(N)
930 FOR X=0 TO N: A$=""
940 : FOR Y=0 TO 20*RND(1)
950 : A$=A$+CHR$(A+B*RND(1))
960 NEXT: A$(X)=A$:A(X)=X: NEXT: RETURN

```

```

10 CLS: PRINT" ## RUND ##": PRINT
20 X=1234567: B=-1/7
30 FOR I=-4 TO 4: C=10*I: PRINT"auf";C: Y=X
40 FOR J=1 TO 12
50 PRINT INT(Y/C+.5)*C,: Y=Y*B
60 NEXT: PRINT: INPUT"weiter";Z
70 NEXT

```

```

## RUND ##
auf 1E-04
1.23457E+06 -176367 25195.2
-3599.32 514.189 -73.4555
10.4936 -1.4991 .2142
-.0306 4.4E-03 -6E-04

```

```

auf 1E-03
1.23457E+06 -176367 25195.2
-3599.32 514.189 -73.456
10.494 -1.499 .214
-.031 4E-03 -1E-03

```

```

auf .01
1.23457E+06 -176367 25195.3
-3599.32 514.19 -73.46
10.49 -1.5 .21
-.03 0 0

```

usw.

```

10 CLS: PRINT"## FRAK87 ##": PRINT:A=0: B=A: C=A: D=A: G=50: E=.01
20 DIM A(15,255): X=A: Y=A: INPUT"Zyklen =";Z
30 INPUT"X1, X2 =";X1,X2:XD=(X2-X1)/320
40 INPUT"Y1, Y2 =";Y1,Y2: YD=(Y2-Y1)/256: Y=Y1-YD
50 FOR I=0 TO 255: X=X1-XD: Y=Y+YD: R=-1
60 FOR J=0 TO 319 STEP 20: Q=0: R=R+1
70 FOR K=0 TO 19: X=X+XD: A=X: B=Y: Q=Q+Q
80 FOR N=1 TO Z
90 C=A*A-B*B-X: D=A*B: D=D+D-Y
100 IF ABS(C-A)>E OR ABS(D-B)>E THEN A=C: B=D: NEXT: NEXT: GOTO 140
110 A=C: B=D: IF ABS(A)+ABS(B)<G THEN NEXT: NEXT: GOTO 140
120 IF 2*INT(N/2)<N THEN Q=Q+1
130 NEXT K
140 A(R,I)=Q: NEXT: PRINT I;: NEXT
150 PRINTX1;X2;Y1;Y2;Z;G;E
160 INPUT"Name";A$: CSAVE*" "+A$;A

```

0 REM: Maschinencode

PRINT, ↓

```

10 WINDOW 0,31,0,39: DIM A(15,255): CLS: PRINT TAB(15);"## FRAK3 ##":
20 INPUT"Name";A$: CLOAD*" "+A$;A: CLS: F=524288
30 FOR X=0 TO 255: D=0
40 FOR Y=0 TO 15: C=A(Y,X): E=F
50 FOR I=0 TO 19: IF C>=E THEN PSET D,X,7: C=-E
60 E=E/2: D=D+1: NEXT:NEXT: NEXT: CALL*406
70 INPUT"";A: IF A<>0 THEN 70

```

```

10 WINDOW 0,31,0,39: CLS: PRINT TAB(12);"## FRAKTAL ##": PRINT
20 DIM A(31,319), B(7): A=0: B=A: C=A: D=A: X=A: Y=A: N=A: G=100: E=.01
30 INPUT"Zyklen =";Z
40 INPUT"X1, X2 =";X1,X2: XD=(X2-X1)/320: X=X1-XD
50 INPUT"Y1, Y2 =";Y1,Y2: YD=(Y2-Y1)/256: CLS
60 !-----Bildzeugung-----
70 FOR I=0 TO 319: Y=Y1-YD: X=X+XD
80 FOR J=0 TO 255 STEP 8
90 FOR K=0 TO 7: Y=Y+YD: A=.5: B=0
100 FOR N=1 TO Z: B(K)=4
110 C=A*A-B*B-X: D=A*B: D=D+D-Y
120 IF ABS(A-C)<E AND ABS(B-D)<E THEN NEXT K: GOTO 160
130 A=C: B=D
140 IF ABS(A) + ABS(B)>G THEN B(K)=N-5*INT(N/5): NEXT K: GOTO 160
150 NEXT: NEXT
160 A=0: FOR K=0 TO 7: A=A*5+B(K): IF B(K)=4 THEN PSET I,3+K,7
170 NEXT: A(J/8,I)=A: !Matrizelement
180 NEXT: NEXT
190 !-----Eckpunkte-----
200 PSET 0,0,7: PSET 319,0,7: PSET 0,255,7: PSET 319,255,7
210 INPUT"";A: IF A<>0 THEN 210
220 !-----Bildauswertung-----
230 B=78125: INPUT"Wert 0-4";X: CLS
240 FOR I=0 TO 319
250 FOR J=0 TO 31: L=8*J: C=B: A=A(J,I)
260 FOR K=0 TO 7
270 D=INT(A/C): A=A-D*C: C=C/5
280 IF D=X THEN PSET I,L+K,7
290 NEXT: NEXT: NEXT: GOTO 200

```

```

10 CLS: PRINT TAB(10)"### KOMBA ###": GOTO 680
20 REM===== Multiplikation =====
30 Y=0: FOR I=0 TO N
40   X=A(I)*A+Y: Y=INT(X/S): A(I)=X-S*Y
50 NEXT: IF Y>0 THEN N=N+1: A(N)=Y
60 IF Y>999 THEN A(N)=Y-S: N=N+1: A(N)=1
70 RETURN
80 REM===== Division =====
90 Y=0: FOR I=N TO 0 STEP -1
100  X=A(I)+S*Y: A(I)=INT(X/A): Y=X-A*A(I)
110 NEXT: IF A(N)=0 THEN N=N-1: IF N<0 THEN N=0
120 IF Y>0 THEN PRINT: PRINT"#### Rest="Y"####"
130 RETURN
140 REM===== Eingabe =====
150 INPUT X: IF X<2 THEN PRINT"zu klein": GOTO 150
160 IF X>999 THEN PRINT"zu gross": GOTO 150
170 IF X-INT(X) THEN PRINT"nicht integer": GOTO 150
180 RETURN
190 REM===== Anzeige =====
200 FOR I=N TO 0 STEP -1: FOR J=0 TO 11
210   C=A(I-J): IF C>99 THEN PRINT C;: GOTO 230
220   C$="000"+MID$(STR$(C),2): PRINT "MID$(C$,LEN(C$)-2)+ " ";
230   IF I-J=0 THEN J=15
240 NEXT: PRINT: NEXT
250 PRINT"Die Zahl hat"; 3*N+LEN(STR$(A(N)))-1; "Stellen"
260 PRINT: RETURN
270 REM===== Lange Eingabe =====
280 PRINT: PRINT"Eingabe nur in 3er-Gruppen"
290 PRINT"Jede Eingabe wird so interpretiert"
300 PRINT"Ende der Eingabe mit negativer Zahl": PRINT
310 FOR I=M TO 0 STEP -1: INPUT A(I)
320   IF A(I)-INT(A(I)) THEN PRINT"nicht integer": GOTO 320
330   IF A(I)>999 THEN PRINT"zu gross": GOTO 320
340   IF A(I)<0 THEN N=M-I-1: I=0
350 NEXT: FOR I=0 TO N: A(I)=A(M-N+I): NEXT: RETURN
360 REM===== Fakultaet =====
370 PRINT"Eingabe n=";: GOSUB 150: N=0: A(0)=1: L=X
380 FOR A=2 TO L: GOSUB 30: NEXT
390 PRINT: PRINT"Die Fakultaet von"; L; "betrægt:": RETURN
400 REM===== Potenz =====
410 PRINT"Fuer x^y Eingabe x=";: GOSUB 150: N=0: A(0)=X: A=X
420 PRINT"y=";: GOSUB 150: L=X
430 FOR J=2 TO L: GOSUB 30: NEXT: PRINT
440 PRINT A; "A"; L; "betrægt:": RETURN
450 REM===== n ueber m =====
460 PRINT"n ueber m; n=";: GOSUB 150: L=X: N=0: A(0)=X
470 PRINT"m=";: GOSUB 150: D=X
480 IF L<X+X THEN D=L-X: IF D 2 GOTO 460
490 FOR J=2 TO D: A=L+1-J: GOSUB 30: A=J: GOSUB 90: NEXT: PRINT
500 PRINT L; "ueber"; D; "bzw. ueber"; L-D; "betrægt:": RETURN
510 REM===== Lange Zahl =====
520 PRINT"zuvor bezueglich langer Zahl"
530 INPUT"0:nichts 1:eingeben 2:anzeigen";A
540 IF A=0 THEN RETURN
550 PRINT: ON A GOSUB 280, 200: RETURN
560 REM===== Multiplikation=====
570 GOSUB 520: PRINT"Multiplikator =";: GOSUB 150: A=X: GOSUB 30
580 PRINT"Multiplikation mit"; A; "ergiebt:": RETURN
590 REM===== Division =====

```

```

590 REM===== Division =====
600 GOSUB 520: PRINT " Divisor =";: GOSUB 150: A=X: GOSUB 90
610 PRINT: PRINT "Division mit"; A; "ergiebt:": RETURN
620 REM===== n!/(n-m)! =====
630 PRINT "Fuer n!/(n-m)!; n=";: GOSUB 150: L=X
640 PRINT "m=";: GOSUB 150: D=L-X: N=0: A(0)=D+1
650 FOR A=D+2 TO L: GOSUB 30: NEXT: PRINT
660 PRINT L; "! /"; D; "! betraegt:": RETURN
670 REM===== Start =====
680 PRINT "H.Voelz"; TAB(32) "28.6.87": PRINT: S=1000
690 INPUT "Maximale Stellenzahl"; M: M=INT((M+2)/3): DIM A(M)
700 PRINT "Funktionen : "
710 P=B: PRINT: PRINT "1:Fakultaet 2:Potenzen 3:n ueber m"
720 INPUT "4:Multiplikt. 5:Division 6:n!/(n-m)!"; B: PRINT
730 ON B GOSUB 370, 410, 460, 570, 600, 630: GOSUB 200: GOTO 700

```

```

10 CLS: PRINT "Eulersche Zahl": PRINT
20 INPUT "Stellenzahl"; S
30 PRINT: PRINT "#### bitte warten ####": PRINT
40 V=3: Z=10: A=S/V+2.9: DIM B(A), C(A): N=1: B(A)=1: C(A)=1
50 U=0: FOR J=1 TO A: Q=B(J)+C(J)+U: U=INT(Q/Z)
60 B(J)=INT((Q/Z-U)*Z+.5): NEXT: N=N+1
70 R=0: FOR J=A TO 1 STEP -1
80 Q=R*Z+C(J): C(J)=INT(Q/N): R=INT(Q-N*C(J)): NEXT
90 IF C(1)>0 OR N=2 GOTO 50
100 FOR J=A TO 2 STEP -1: PRINT B(J);: NEXT
110 !      1      1      1      1
120 ! e = 1 + -- + -- + -- + -- + ...
130 !      1!    2!    3!    4!

```

Eulersche Zahl

Stellenzahl 30

bitte warten

2 718 281 828 459 45 235 360 287 471 352 2 718 281

```

10 CLS: PRINT "### FAKUL ###": PRINT
20 INPUT "N ="; N
30 X=LN(8*N*ATN(1))/2+N*LN(N)+N*LN(EXP(-1))
40 X=X/LN(10): Y=INT(X): X=X-Y: X=10^X
50 PRINT X; "E+"; Y
60 GOTO 20

```

FAKUL

N = 127.
3.01081 E+ 213
N = 4576
1.91095 E+ 14765
N = 1E+29
1 E+ 2.85657E+30

```

10 CLS: PRINT TAB(12); "### PI ###": PRINT: GOTO 130
20 !----- Subroutinen -----
30 R=0: FOR J=A TO 1 STEP -1: Q=R*Z+A(J)
40 A(J)=INT(Q/N): R=INT(Q-N*A(J)): NEXT: RETURN
50 R=0: FOR J=A TO 1 STEP -1: Q=Z*R+B(J)
60 B(J)=INT(Q/N): R=INT(Q-N*B(J)): NEXT: RETURN
70 N=25: GOSUB 30: N=2*I+1: GOSUB 30: GOSUB 30: N=57121: GOSUB 50
80 F=2*I-1: U=0: FOR J=1 TO A: Q=A(J)*F
90 W=INT(Q/Z): A(J)=INT((Q/Z-W)*Z+.5)+U: U=W: NEXT
100 FOR J=1 TO A: Q=B(J)*F: W=INT(Q/Z)
110 B(J)=INT((Q/Z-W)*Z+.5)+U: U=W: NEXT: RETURN
120 !----- MAIN -----
130 INPUT "Stellenzahl"; S
140 PRINT: PRINT "### bitte warten ###": PRINT
150 V=2: Z=10^V: A=S/V+3.9: DIMA(A), B(A), C(A)
160 A(A)=16: B(A)=4: I=1: N=5: GOSUB 30: N=239: GOSUB 50
170 U=0: FOR J=1 TO A: Q=C(J)+A(J)-B(J)+U: U=INT(Q/Z)
180 C(J)=INT((Q/Z-U)*Z+.5): NEXT: GOSUB 70: I=I+1
190 U=0: FOR J=1 TO A: Q=C(J)-A(J)+B(J)+U: U=INT(Q/Z)
200 C(J)=INT((Q/Z-U)*Z+.5): NEXT
210 GOSUB 70: I=I+1: IF A(I)>0 OR N=3 GOTO 170
220 FOR J=A TO 3 STEP -1: PRINT C(J);: NEXT
230 ! Formel von Gregory 1671:
240 !      1      1      1
250 !PI/4=ATN(1)= 1 - - + - - - +---...
260 !      3      5      7
270 ! Formel von J.Machim 1706
280 !      1      1
290 ! ATN(1) = 4*ATN(-) - ATN(---)
300 !      5      239

```

PI

Stellenahl 20

bitte warten

3 14 15 92 65 35 89 79 32 38 46

```

10 CLS: PRINT TAB(12); "## TEFRA ##": PRINT
20 H=199: A=0: B=0: M=0: X=0: Y=0: Q=23: IF PEEK(-5) 130 THEN Q=31: H=155
30 INPUT "X1, X2 ="; X1, X2: XD=(X2-X1)/40: Z=27: G=50: E=.01
40 INPUT "Y1, Y2 ="; Y1, Y2: YD=(Y2-Y1)/Q: Y=Y2+YD: WINDOW 0,Q,0,39: CLS
50 FOR I=0 TO Q-1: X=X1-XD: Y=Y-YD
60 FOR J=0 TO 3: X=X+XD: A=.5: B=0: C=A: D=B: F=H
70 FOR N=1 TO Z
80 C=A*A-B*B-X: D=A*B: D=D+D-Y
90 IF ABS(C-A)<E AND ABS(D-B)<E THEN F=N+95: N=Z
100 A=C: B=D: IF ABS(A) + ABS(B) > G THEN F=N+63: N=Z
110 NEXT: PRINT CHR$(F);: NEXT: NEXT
120 INPUT "": A: IF A=0 THEN END

```

```

0 rem: Maschinencode
10 CLEAR 30,6100: WINDOW 0,31,0,39: DIM S(320): A=0: B=A: C=A: D=A: GOTO
20 !-----Iteration-----
30 FOR N=1 TO Z
40 C=A*A-B*B-X: D=A*B: D=D+D-Y
50 IF ABS(C-A)<E AND ABS(D-B)<E THEN N=Z
60 A=C: B=D: IF ABS(A)+ABS(B) > G THEN H=N/2: N=Z
70 NEXT: RETURN
80 !-----Main-----
90 E=.01: Y=A: G=100: H=A: N=A: CLS: PRINT TAB(15)"## FRAGEB ##": PRINT
100 INPUT"Zklen = ";Z
110 INPUT"X1, X2 ";X1, X2: XD=(X2-X1)/200
120 INPUT"Y1, Y2 ";Y1, Y2: YD=(Y2-Y1)/240: YG=YD*8
130 INPUT"1: Gebirge 2: Fraktal";A: ON A GOSUB 210, 150: GOTO 130
140 !-----Fraktal-----
150 Y=Y1-YD: CLS
160 FOR I=0 TO 239: Y=Y+YD: X=X1-XD
170 FOR J=0 TO 199: X=X+XD: A=.5: B=0: H=1
180 GOSUB 30: IF H=INT(H) THEN PSET J,I,7
190 NEXT: NEXT: CALL*406: INPUT"";A: IF A=0 THEN RETURN
200 !-----Gebirge-----
210 Y=Y1-YG: FOR I=0 TO 320: S(I)=0: NEXT: CLS
220 FOR I=0 TO 120 STEP 4: Y=Y+YG: X=X1-XD
230 FOR J=0 TO 199: X=X+XD: A=.5: B=0: H=Z: Q=S(I+J)
240 IF Q<S(I+J+1) THEN Q=S(I+J+1)
250 Q=Q+1: GOSUB30: IF I+H>Q THEN S(I+J)=I+H: PSET I+J,I+H,7
260 IF J=0 THEN F=H: NEXT: NEXT: GOTO 310
270 IF ABS(F-H)<2 THEN F=H: NEXT: NEXT: GOTO 310
280 A=F: B=H: IF H<F THEN A=H: B=F
290 FOR M=A+1 TO B-1: IF I+M>Q THEN PSET I+J,I+M,7
300 NEXT: F=H: NEXT: NEXT
310 CALL*406: INPUT"";A: IF A=0 THEN RETURN

```

```

0 REM: Maschinencode
10 CLEAR 30,6100: WINDOW 0,31,0,39: CLS: PRINT TAB(13)"## FRAKI ##":
20 A=0: B=A: C=A: D=A: Y=A: N=A: G=100: E=.01: H=A
30 INPUT"Zyklen = ";Z
40 INPUT"X1, X2 = ";X1, X2: XD=(X2-X1)/320: X=X1-XD
50 INPUT"Y1, Y2 = ";Y1, Y2: YD=(Y2-Y1)/256: CLS
60 FOR I=0 TO 319: Y=Y1-YD: X=X+XD
70 FOR J=0 TO 255: Y=Y+YD: A=.5: B=0: H=.5
80 FOR N=1 TO Z
90 C=A*A-B*B-X: D=A*B: D=D+D-Y
100 IF ABS(C-A)<E AND ABS(D-B)<E THEN N=Z
110 A=C: B=D: IF ABS(A)+ABS(B) > G THEN H=N/2: N=Z
120 NEXT: IF H>INT(H) THEN PSET I,J,7
130 NEXT: NEXT: CALL*406
140 INPUT"";A: IF A<>0 THEN 140

```

```

10 CLS: PRINT"    ##  REFOR  ##""
20 X=2.1E8: B=-1/7
30 FOR I=1 TO 15
40 PRINT TAB(20-LEN(STR$(X)));X: X=X*B
50 NEXT

```

```

##  REFOR  ##
      2.1E+08
      -3E+07
      4.28572E+06
      -612245
      87463.6
      -12494.8
      1784.97
      -254.996
      36.428
      -5.204
      .743428
      -.106204
      .015172
      -2.16743E-03
      3.09633E-04

```

```

10 CLS: PRINT"    ##  PUFOR  ##""
20 X=1.23E8: B=-1/8
30 FOR I=1 TO 15: X$=STR$(X): L=LEN(X$)
40   FOR J=1 TO L
50     IF MID$(X$,J,1)="." THEN L=J-1: J=15
60   NEXT: PRINT TAB(10-L);X: X=X*B
70 NEXT

```

```

##  PUFOR  ##
      1.23E+08
      -1.5375E+07
      1.92188E+06
-240234
  30029.3
-3753.66
  469.208
-58.651
  7.33137
- .916422
  .114553
- .0143191
  1.78989E-03
-2.23736E-04
  2.7967E-05

```

```

220 YV=(Y-YU)*YF:C=C-1:A=C:REM18
230 Q=Y*Y:REM+Y*Y):REM:.....:
240 FORX=XUTOXOSTEPXW:A=A+1:REM8
250 Z=SQR(Q+X*X): Z=COS(Z)*EXP(A1*Z):REMREM:.....:
260 S=U: U=UF+(X-XU)*XF-YV: T=V: W=V: V=(Z-ZU)*ZF+YV
270 !----- Entscheidungen -----
280 G=F: F=2: IF V>G(1,A) AND G(1,A) THEN F=4
290 IF V>254 THEN F=5
300 IF V<G(1,A) AND V G(0,A) THEN F=3
310 IF V<1 THEN F=1
320 IF X=XU AND F=2 THEN XS=U: YS=V: NEXT: NEXT: GOTO 690
330 IF X=XU THEN: NEXT: NEXT: GOTO 690
340 H=0: ON G GOTO 350, 450, 500, 520, 400
350 IF F>1 THEN W0=1: GOSUB 90:IF F=2 THEN XZ=U: YZ=V: GOSUB 20
360 IF F>2 THEN B=0: GOSUB 120: XZ=S: YZ=T: GOSUB 20
370 IF F>3 THEN B=1: GOSUB 120: XS=S: YS=T: IF F=4 THEN XZ=U: YZ=V:
380 IF F=5 THEN W0=254: GOSUB 100
390 GOTO 150
400 IF F<5 THEN W0=254: GOSUB 90: IF F=4 THEN XZ=U: YZ=V: GOSUB 20
410 IF F<4 THEN B=1: GOSUB 120: XZ=S: YZ=T: GOSUB 20
420 IF F<3 THEN B=0: GOSUB 120: XS=S: YS=T: IF F=2 THEN XZ=U: YZ=V:
430 IF F<2 THEN W0=1: GOSUB 100
440 GOTO 150
450 IF X=XU+XW AND F<5 AND F<1 THEN XZ=U: YZ=V: GOSUB 20:GOTO 150
460 IF F=2 THEN XZ=U: YZ=V: GOSUB 20: GOTO 150
470 IF X=XU+XW AND F 3 GOTO 520
480 IF F>2 GOTO 360
490 GOTO 430
500 IF F>2 GOTO 370
510 IF F<3 GOTO 420
520 IF F=4 THEN XZ=U: YZ=V: GOSUB 20: GOTO 150
530 IF F=5 GOTO 380
540 GOTO 410
550 !----- Start -----
560 WINDOW 0,31,0,39: COLOR7,0: CLS
570 PRINT TAB(10)"### PLOT3D ###"
580 PRINT"H.VOELZ"TAB(30)"85/87": PRINT: DIM G(1,75)
590 !----- Eingaben -----
600 A$="": INPUT"Def.Konst.A1-AZ":A$: IF A$<>" THEN CALL*40A
610 A$="": INPUT"Hilfsfkt. Q(Y)=":A$: IF A$<>" THEN CALL*40F
620 A$="": INPUT"Z(X,Y,Konst.)=":A$: IF A$<>" THEN CALL*414
630 INPUT"Linienzahl=":L: IF L<2 THEN L=2
640 IF L>25 THEN L=25
650 R=L+L:INPUT"X-Bereich=":XU,X0: UF=106: XF=212/(X0-XU): XW=(X0-XU)/R
660 INPUT"Y-Bereich=":YU,Y0: YF=105/(Y0-YU)
670 INPUT"Z-Bereich=":ZU,Z0: ZF=170/(Z0-ZU)
680 FOR X=0 TO 75: G(0,X)=999: G(1,X)=-99: NEXT: GOTO 190
690 INPUT"":A$: IF A$<>"0" GOTO 690
700 PRINT XU: X0: YU: Y0: ZU: Z0
710 INPUT"":A$: IF A$<>"0" GOTO 690
720 GOTO 600

```

```

780 REM#### FILE #####
790 REM----- ZAHLEN -----
800 INPUT"Startwert >0 ";A: A=RND(-A)
810 INPUT"Zahlenbereich";A,B: B=B-A
820 INPUT"  Anzahl  ";N: DIM AZ(N),A(N)
830 FOR X=0 TO N: AZ(X)=A+B*RND(1):A(X)=X: NEXT: RETURN
890 REM----- TEXTE -----
900 INPUT"Startwert >0 ";A: A=RND(-A)
910 CLEAR 10000: A=ASC("A"): B=ASC("Z")-A+1
920 INPUT"  Anzahl  ";N: DIM A$(N),A(N)
930 FOR X=0 TO N: A$="": A(X)=X
940 : FOR Y=0 TO 20*RND(1)
950 : A$=A$+CHR$(A+B*RND(1))
960 NEXT: A$(X)=A$: NEXT: RETURN

```

```

10 PRINT TAB(10);"## ZABUB ##": PRINT
20 GOSUB 800: INPUT"Start";A
30 FOR X=0 TO N: F=0: FOR Y=0 TO N-1
40 IF AZ(Y)>AZ(Y+1) THEN A=AZ(Y): AZ(Y)=AZ(Y+1): AZ(Y+1)=A: F=1
50 NEXT: PRINT X;: IF F=0 THEN X=N
60 NEXT
70 FOR X=0 TO N: PRINT AZ(X);: NEXT: END
790 REM----- ZAHLEN -----
800 INPUT"Startwert > 0 ";A: A=RND(-A)
810 INPUT"Zahlenbereich";A,B: B=B-A
820 INPUT"  Anzahl  ";N: DIM AZ(N)
830 FOR X=0 TO N: AZ(X)=A+B*RND(1): NEXT: RETURN

```

```

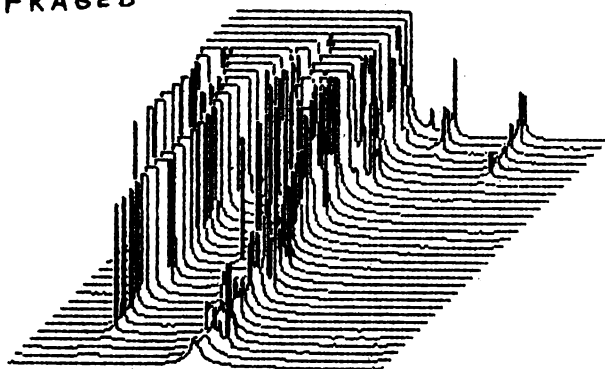
10 PRINT TAB(10);"## WOBUB ##": PRINT
20 CLEAR 10000: GOSUB 900: INPUT"Start";A
30 FOR X=0 TO N: F=0: FOR Y=0 TO N-1
40 IF A$(Y)>A$(Y+1) THEN A=A$(Y): A$(Y)=A$(Y+1): A$(Y+1)=A: F=1
50 NEXT:PRINT X;: IF F=0 THEN X=N
60 NEXT
70 FOR X=0 TO N: PRINT A$(X): NEXT: END
890 REM----- TEXTE -----
900 INPUT"Startwert >0 ";A: A=RND(-A)
910 A=ASC("A"): B=ASC("Z")-A+1
920 INPUT"  Anzahl  ";N: DIM A$(N), A(N)
930 FOR X=0 TO N: A$=" "
940 : FOR Y=0 TO 20*RND(1)
950 : A$=A$+CHR$(A+B*RND(1))
960 NEXT: A$(X)=A$: NEXT: RETURN

```

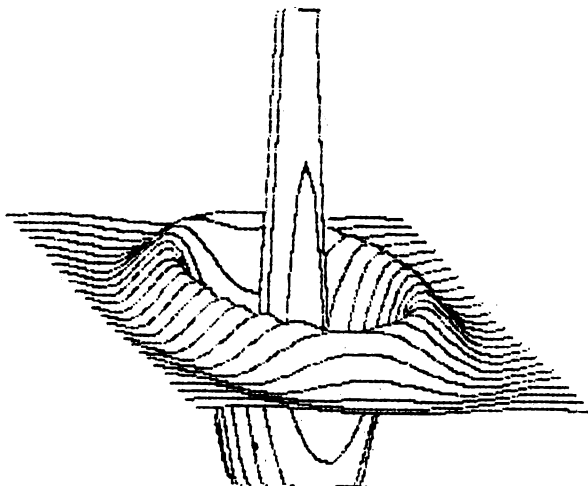
[illegible]



FRAGEB



PLDT30



BASUHR

```

0400 00 E3 04 00 00 0E F3 F5 _C 00 00 su
0400 00 01 10 04 22 00 01 0E 01 00 00 00 00
0410 00 07 03 0C 0E 09 03 0C 01 00 00 00 00
0410 00 F1 F0 C9 F3 C5 05 05 F5 00 01 0E 00
0420 00 00 00 C0 07 03 00 0A 04 00 00 00 00
0420 00 01 C0 02 27 32 04 01 30 00 00 00 00
0430 00 3F 3A 03 01 C6 01 27 32 00 00 00 00
0430 00 03 01 06 00 20 23 32 03 00 00 00 00
0440 00 01 3A 02 01 C6 01 27 32 00 00 00 00
0440 00 02 01 06 00 20 13 32 02 00 00 00 00
0450 00 01 3A 01 01 C6 01 27 32 00 00 00 00
0450 00 01 01 06 24 20 03 32 01 00 00 00 00
0460 00 01 2A 00 07 22 05 01 0F 00 00 00 00
0460 00 67 2E 19 22 00 07 21 01 00 00 00 00
0470 00 01 7E 23 CD 03 F0 1C CD 00 00 00 00
0470 00 03 F0 20 7E 23 CD 03 F0 00 00 00 00
0480 00 1C CD 03 F0 20 7E CD 03 00 00 00 00
0480 00 F0 1C 2A 05 01 22 00 07 00 00 00 00
0490 00 0E 03 03 0C F1 C0 F3 F5 00 00 00 00
04A0 00 05 05 C5 00 00 C0 07 03 00 00 00 00
04A0 00 01 01 01 06 04 7E CD 00 00 00 00 00
04B0 00 03 F0 1C CD 03 F0 20 23 00 00 00 00
04B0 00 10 F4 CD 03 F0 2C C1 D1 00 00 00 00
04C0 00 E1 F1 F0 C9 0C 0A 43 6F 00 00 00 00
04C0 00 70 79 72 69 67 60 74 20 00 00 00 00
04D0 00 40 2E 56 6F 65 6C 7A 20 00 00 00 00
04D0 00 44 65 7A 2E 31 39 38 37 00 00 00 00
04E0 00 0A 00 00 FC 04 0A 00 94 00 00 00 00
%

```

BASUHR ## 22 41 34

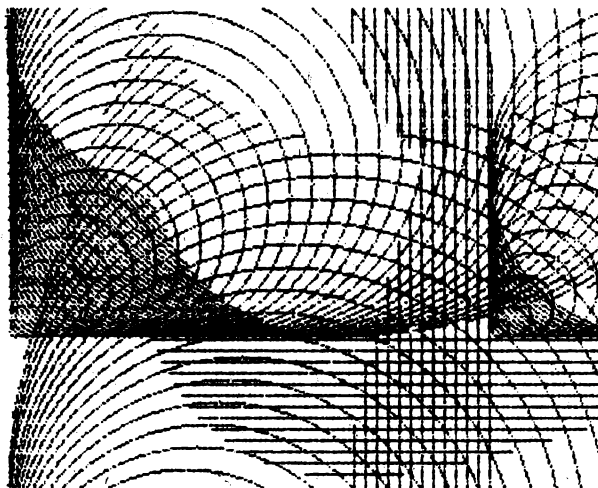
```

1: ein- 2: ausschalten 3: Stoppuhr 1
h,m,s = 22,40,10
1: ein- 2: ausschalten 3: Stoppuhr 3
Zyklen= 200
22 40 10 44
22 40 10 86
1: ein- 2: ausschalten 3: Stoppuhr 3
Zyklen= 10000
22 40 25 64
22 40 45 42
1: ein- 2: ausschalten 3: Stoppuhr 3
Zyklen= 10
22 40 55 36
22 40 55 42

```

%DISPLAY 35F 365
035F 57 04 00 9B 00 00 00 00 WEL

COPY



```

10 CLS: PRINT TAB(9); "**** BINOM ****": PRINT
20 PRINT: INPUT "N,M"; N,M: A=N
30 IF M=1 THEN 50
40 FOR X=2 TO M: A=A*(N-X+1)/X: NEXT
50 PRINT "N ueber M ="; A: GOTO 20

```

```

10 CLS: PRINT TAB(9); "**** FIBON ****": PRINT
20 PRINT: INPUT "Ordnung ="; N: IF N>INT(N) OR N<0 THEN 20
30 A=1: B=2: IF N=1 THEN PRINT A: GOTO 20
40 IF N=2 THEN PRINT B: GOTO 20
50 FOR I=3 TO N: C=A+B: A=B: B=C: NEXT
60 PRINT C: GOTO 20

```

```

10 CLS: PRINT TAB(9); "**** GGT ****": PRINT
20 PRINT: INPUT "N,M"; N,M
30 FOR I=0 TO 1 STEP 0
40 IF M=0 THEN PRINT "GGT ="; N: GOTO 20
50 IF M>N THEN A=M: M=N: N=A
60 R=N-M*INT(N/M): N=M: M=R
70 NEXT

```

```

10 CLS: PRINT TAB(9); "**** ACK1 ****": PRINT
20 INPUT "Ymax ="; N: DIM A(4,N)
30 !=== Zeile 0 belegen ===
40 FOR Y=0 TO N: A(0,Y)=Y+1: NEXT
50 !=== Umspeicherung fuer Y=0 ===
60 FOR X=1 TO 4: A(X,0)=A(X-1,1)
70 FOR Y=1 TO N-1
80 IF A(X,Y-1)>N THEN NEXT X: GOTO 130: ! zu grosses Argument
90 IF A(X-1,A(X,Y-1))=0 THEN NEXT X: GOTO 130: ! nicht berechnet
100 A(X,Y)=A(X-1,A(X,Y-1))
110 NEXT: NEXT
120 !===== Anzeige =====
130 FOR Y=0 TO N: M=(Y+1)/10: FOR X=0 TO 4
140 PRINT TAB(X*6); A(X,Y);
150 NEXT: PRINT: IF M=INT(M) THEN INPUT ""; A
160 NEXT

```

```

10 CLS: PRINT TAB(9); "**** ACK2 ****": PRINT: Q=1
20 INPUT "Ymax ="; N: DIM S(N)
30 INPUT "X,Y"; S(0),S(1): T=1
40 IF T=0 THEN PRINT: PRINT "Ergebnis ="; S(0): GOTO 30
50 IF S(T-1)=0 THEN S(T-1)=S(T)+1: T=T-1: GOTO 80
60 IF S(T)=0 THEN S(T-1)=S(T-1)-1: S(T)=1: GOTO 80
70 S(T+1)=S(T)-1: S(T)=S(T-1): S(T-1)=S(T-1)-1: T=T+1
80 FOR X=0 TO T: PRINT S(X);: NEXT: Q=Q+1: PRINT: ! Stackausdruck
90 IF Q=20 THEN INPUT ""; A: Q=1
100 GOTO 40

```

Ergebnis = 7

Sie empfangen uns mittwochs um 17.00 Uhr im Schulfunk
von Radio DDR II und am Sonnabend im Computerclub
von Jugendlradio DT 64

Sendetag		Thema	Inhalt
Schulfunk	DT 64		
13.1.	16.1.	1	Sortieralgorithmen
10.2.	13.2.	2	Formatierung, Kombinatorik
9.3.	12.3.	3	Fraktale, chaotische Probleme
13.4.	16.4.	4	Einbinden von Maschinencode
11.5.	14.5.	5	Rekursive Programmierung
8.6.	11.6.	6	Analytische Methoden, Formel- manipulation
14.9.	17.9.	7	Lernfähige Programme
12.10.	15.10.	8	Menu- und Eingabetechniken
9.11.	12.11.	9	PEEK, DEEK, POKE, DOKE, USR
14.12.	17.12.	10	Fitten von Daten

Hinweis: Außer "BASIC für Fortgeschrittene" senden wir
am vierten Mittwoch/Sonnabend den
" Software - Service "
und am ersten und dritten Mittwoch/Sonnabend
des Monats wird "BASIC - 1x1 des Programmierens"
wiederholt.

Unsere Sendefrequenzen auf UKW :

	Radio DDR II	DT 64
Frankfurt(O.)	87,6 MHz	101,5 MHz
Putbus	88,6 MHz	91,5 MHz
Helpterberg	90,5 MHz	103,8 MHz
Marlow	91,0 MHz	100,8 MHz
Sonneberg	91,7 MHz	102,7 MHz
Dresden	92,2 MHz	102,4 MHz
Inselsberg	92,5 MHz	102,2 MHz
Schwerin	92,8 MHz	101,3 MHz
K.-M.-Stadt	92,8 MHz	100,0 MHz
Suhl	93,7 MHz	/
Leipzig	93,9 MHz	102,9 MHz
Brocken	94,6 MHz	101,4 MHz
Dequede	94,9 MHz	101,0 MHz
Löbau	98,2 MHz	91,8 MHz
Cottbus	98,6 MHz	103,2 MHz
Berlin	99,7 MHz	102,6 MHz
		und 93,1 MHz
Hoyerswerda	100,4 MHz	/
Marlow	102,8 MHz	/

sowie auf MW :

Burg	/	657 kHz
Neubrandenburg	/	657 kHz
Reichenbach	/	657 kHz

Dieses Heft entstand in Zusammenarbeit von Radio DDR II mit Prof.Dr.Horst Völz und der Abteilung Öffentlichkeitsarbeit des Rundfunks der DDR.

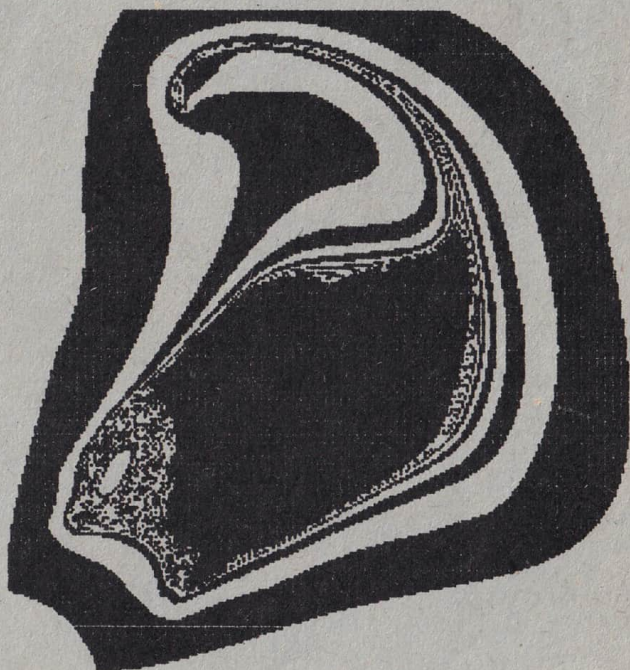
Teil 2

basic

für

Fortgeschrittene

Autor: Prof.Dr.H.Völz



Sendung 6

1. Das Vierfarbenproblem kann beispielhaft an einer politischen Landkarte betrachtet werden. Die Frage besteht dann darin, ob es bei beliebiger Anordnung der Länder immer möglich ist, mit 4 Farben auszukommen, ohne daß sich gleiche Farben an einer Grenze berühren.

2. Hierbei konnte gezeigt werden, wie die Rechentechnik auch bei mathematischen Beweisen wirksam werden kann. Sie ermöglicht nämlich eine weitaus größere Anzahl von Fällen zu durchmustern. Die Anzahl dieser Fälle muß aber endlich sein.

3. Wird die Anzahl der Fälle extrem groß, so übersteigt die Rechenzeit alle Grenzen und das Problem ist damit nicht mehr mittels der Rechentechnik lösbar.

4. Werden die Zahlenwerte sehr groß, oder müssen gar unendlich große Zahlen, wie beim Hydraproblem zugelassen werden, so kann ebenfalls nur ein kleiner Teil der Fälle durchmustert werden. Dennoch hilft in diesen Fällen die Rechentechnik etwas in dem Sinne, daß sie die Vermutung weiter unterstützt. Der eigentliche Beweis muß jedoch mit den bekannten mathematischen Mitteln erbracht werden.

5. Seit einigen Jahren sind Programmpakete bekannt, mit denen man formal wie in der Analytischen Mathematik arbeiten kann. Es können also Formeln vereinfacht, abgeleitet oder integriert werden und vieles andere mehr. Es stehen schon jetzt für Kleinstrechner einfache Methoden bereit. Damit gelangt die Rechentechnik auf ein neues Niveau, nämlich von der Rechnung mit Zahlenwerten zum Umgang mit Formeln.

Sendung 7

1. Die Künstliche Intelligenz läßt sich als ein wissenschaftliches Arbeitsgebiet beschreiben, das heute vor allem zwei Ziele verfolgt:

- # Die Entwicklung neuer hochleistungsfähiger Rechner

- # Simulation von intelligentem menschlichen Verhalten mittels der Methoden der Rechentechnik.

Im zweiten Fall ist dabei nicht entscheidend oder gefordert,

daß gleiche oder ähnliche Ergebnisse auch gleiche Algorithmen oder Denkprinzipien bedeuten.

2. Einige Probleme lassen sich über die Durchmusterung aller möglichen Kombinationen lösen. Hierzu gehört z.B. das Wasserglasproblem. Doch so etwas ist die Ausnahme.

3. Eine wichtige Methode der Künstlichen Intelligenz ist das backtracking. Hierbei werden alle vorangegangenen Züge auf einen Stack gespeichert. Falls man sich geirrt hat, so ist der gegangene Weg gespeichert und es können die Züge teilweise zurückgenommen und durch andere ersetzt werden. Eine solche Methode ist beim Labyrinth am Beispiel des Ariadne-Fadens bekannt. Das Damenproblem gehört ebenfalls zu dieser Problemklasse.

4. In einigen Fällen ist es möglich, Spiele lernfähig zu gestalten. In einer Matrix führen dabei die Erfolgszüge zu höherer und die Verlustzüge zu niedrigerer Bewertung. Die Zugwahl erfolgt dann neben dem Zufall durch die Größe dieser Koeffizienten. So kann mit einem an sich leistungsschwachen Programm begonnen werden, welches aber über die Änderung dieser Koeffizienten ständig seine Leistungsfähigkeit steigert.

5. Für komplizierte Probleme kann man weder eine solche Matrix hinreichend groß machen, noch kann man durch Kombinatorik die besten Züge ausrechnen. Dann sind komplizierte, heuristische Algorithmen erforderlich. Sie werden u.a. bei Schachprogrammen angewendet. U.a. werden dabei je nach Spielsituation und Figurenstärke die Entscheidungen bezüglich der Zugwahl verändert.

Sendung 8

1. Die Menus sollten kurz und eindeutig sein. Die Praxis zeigt, daß zu lange Texte meist nicht gelesen werden.

2. Die Menus sollten auch gewissen ästhetischen Ansprüchen genügen. Wenn man die Möglichkeit hat, Sonderzeichen zu generieren, wie z.B. beim KC 85/2 und 3 sollte davon Gebrauch gemacht werden.

3. Für die Antworten sollten nicht zu viele Wahlmöglichkeiten angeboten werden. Ein Optimum liegt zwischen 3 und 8. Sind mehr notwendig, so ist es besser, Untermenüs zu bilden.

4. Bei der Eingabe sollte jedes Programm so gestaltet sein, daß keine Eingabe zu einer fehlerhaften Reaktion des Rechners oder gar zum Absturz führt. Diese Forderung ist sehr schwer zu realisieren, da in der Regel die volle Kombinatorik aller Eingaben schwer zu überblicken ist. Dennoch sollte der Programmierer hierzu sehr große Sorgfalt walten lassen.

5. Die Cursor-bezogene Eingabe ist in BASIC schwer* zu realisieren. Sie hat aber drei wichtige Vorteile:

- # Es werden Fehleingaben nahezu vollständig unterbunden

- # Die Menu-Gestaltung läßt sich in besonders übersichtlicher Form realisieren.

- # Sie führt zu modernen Eingabetechniken, die z.B. durch eine Maus oder Joystick gesteuert werden.

SENDUNG 9

1. Maschinennahe Befehle ermöglichen den unmittelbaren Zugriff auf die Ressourcen des Rechners. Hierbei können folgende Befehlsgruppen unterschieden werden:

- # unmittelbare, wie CLEAR, LINES, RESTORE, WIDTH usw.

- # mit direktem Zugriff auf den Arbeitsspeicher, wie

 - PEEK, DEEK, POKE, DOKE, VPEEK UND VPKE

- # mit direktem Zugriff auf die Peripherie, wie INP, OUT und WAIT

- # das Ausnutzen von Maschinenroutinen über CALL und

- # der leistungsfähige Befehl USR

2. Diese Befehle besitzen Vor- und Nachteile. Die Vorteile bestehen vor allem in der höheren Geschwindigkeit. Die Nachteile betreffen u.a.:

- # die Übertragbarkeit auf andere Rechner

- # notwendige Kenntnisse des Maschinencodes

- # Gefahr des Systemabsturzes

- # Wahl eines günstigen Speicherbereiches

3. Bei einem fertigen Programm kommen jedoch fast nur die Vorteile zum Tragen. Deshalb sollten solche Programme so geschrieben sein, daß der Nutzer nicht das Maschinenprogramm merkt.

4. Bei der Entwicklung eines kombinierten BASIC-Maschinen-Programms sollten öfter als sonst üblich, das Programm und Teilroutinen gerettet und bis zur Fertigstellung aufgehoben

werden.

5. Der Speicherzugriff mit maschinennahen Befehlen auf Bereiche innerhalb des Programms, der Variablen usw. muß mit größter Umsicht erfolgen.
6. In den unbenutzten Bereichen sind Daten mit hoher Dichte unter anderem durch POKE und DOKE abzulegen und mittels PEEK und DEEK zurückzuholen.
7. Der Speicherraum zwischen den Feldern und dem Stack sollte immer erst dann bestimmt werden, wenn bereits alle benutzten Variablen und Felder initialisiert sind.
8. Auch in BASIC existiert die Methode, ein Programm sich selbst modifizieren zu lassen. In Sonderfällen bringt diese verpönte Methode Vorteile.
9. Der Zugriff auf Ports erfolgt mit den Befehlen INP und OUT. Dies setzt genaue Kenntnisse bezüglich der Peripherie des Rechners voraus.
10. Die USR-Funktion ist eine CALL-Variante mit zusätzlicher Übergabe von Werten. Bei A=USR(X) wird der Wert von X an das DE-Register übergeben. Dies realisiert eine Routine des BASIC-Interpreters auf der Adresse C96F. Die Rückgabe von Ergebnissen kann auf die Variable A aus den Registern A und B erfolgen. Die zugehörige Routine liegt auf der Adresse D0B1. Die Startadresse der Maschinenroutine muß mit DOKE 772,Adr. eingetragen werden.

SENDUNG 10

1. Meßwerte sind stets mit Fehlern behaftet. Durch wiederholte Messung bewirken die zufälligen Fehler eine Streuung um den richtigen Mittelwert. So ist es möglich mit mehreren Messungen dem wirklichen Wert näher zu kommen oder ihn einzugrenzen.
2. Mit statistischen Methoden ist es möglich, Aussagen über die vermutliche Größe des Fehlers zu machen. Ein Maß hierfür ist die Streuung. Sie kann aus mehreren Meßwerten berechnet werden.
3. Es gibt verschiedene statistische Maße, welche Hinweise auf Eigenschaften von Meßwerten und Fehlern geben.
4. Bei mehreren Anwendungen ändert sich eine Ausgangsgröße

unter dem Einfluß der Eingangsgröße. Wichtig ist es hierbei den Zusammenhang zu bestimmen. In vielen Fällen besteht direkte Proportionalität, die aber durch Störungen verdeckt wird. Ein Maß für den Grad der Gültigkeit einer solchen Annahme ist die Korrelation. Sowohl der Proportionalitätsfaktor als auch der Korrelationskoeffizient sind gut zu berechnen.

5. Neben dem linearen Zusammenhang existieren noch viele andere. Besonders wichtig sind der reziproke, exponentielle und jener mit einer Potenz.

6. Für den Zusammenhang von Eingangs- und Ausgangsgrößen sind zweiparametrische Funktionen von großer Bedeutung. Der meist verwendete Ansatz geht von der Beziehung $Y=A+B \cdot X$ aus. Nach der Methode der kleinsten quadratischen Abweichung sind die Parameter A und B sowie der Korrelationskoeffizient zu berechnen. Dieses Prinzip heißt im englischen FIT. Im deutschen ist Anpassung gebräuchlich. Nicht selten wird auch von Regression gesprochen. Dieses beinhaltet aber bereits eine zusätzliche Anwendung, nämlich mit den berechneten Parametern auf Werte zu schließen, die außerhalb der Daten liegen.

7. Neben dem linearen FIT gibt es auch logarithmische, exponentielle usw. Sie sind jedoch nicht unmittelbar zu realisieren. Es muß zusätzlich die Linearisierung der Funktion durchgeführt werden.

Korrektur zu Programmen des Teil 1:

Programm	Zeilennummer
	46 ... IF N<S THEN PRINT
MINTEX	30 IF J<=30 THEN
	38 ... ;:IFS>M+N THEN PRINT " ";
	64 IF J<=THEN
	78 IFS\$<>"\$\$"THEN
	106:IFN<XTHENPRINT#2"
	108 IF I/Y>INT(I/Y)THENRETURN:...
QUICK	90 IF A\$(A(J))<=A\$(H).THEN 120
KOMBA	480 ...: IFD<2 GOTO 460
TEFRA	20 IF PEEK(-5)<130 then Q=31:H=155
KOMBA	280 ... AND G(1,A)>G(0,A).THEN F=4
	300 ... AND V G(0,A)>THEN F=3
	450 ... AND F<5 AND F>1 THEN XZ=U: ...

```

10 PRINT TAB(10)"**** PRIMEL ****": PRINT: GOTO 70
20 REM ----- TEST PRIMZahl FUER P, Q=FLAG -----
30 Q=0: FOR X=3 TO SQR(P+1) STEP 2: T=P/X
40 IF T=INT(T) THEN X=P: Q=1
50 NEXT X: RETURN
60 REM ----- MAIN -----
70 INPUT " Ende = ";E: M=3E7
80 INPUT "Abstand = ";A: IF A/2<>INT(A/2) OR A<=2 GOTO 80
90 PRINT: PRINT "Zahl", "Zyklen", "Maximum": PRINT
100 FOR Y=3 TO E STEP 2: P=Y: D=Y: Z=0: GOSUB 30
110 IF Q=1 THEN NEXT Y: END
120 P=P*P+A: Z=Z+1: IF P>D THEN D=P
130 IF P>M THEN PRINT Y;"Grenze",Z,P: NEXT Y: END
140 GOSUB 30: IF Q=0 GOTO 120
150 IF Z>1 THEN PRINT Y,Z,D
160 NEXT Y: END

```

```

10 PRINT TAB(12);"## VIQUA ##": PRINT
20 DEF FN A(X)=INT (SQR(X))
30 INPUT "Anfang , Ende";A,B
40 FOR X=A TO B: Y=X/4
50 PRINT "### ";X;" ###"
60 FOR I=FN A(X) TO FN A(Y) STEP -1
70 X1=X-I*I: IF X1=0 THEN 200
80 Y=X1/3: A=FN A(X1): IF A>I THEN A=I
90 FOR J=A TO FN A(Y) STEP -1
100 X2=X1-J*J: IF X2=0 THEN 190
110 Y=X2/2: A=FN A(X2): IF A>J THEN A=J
120 FOR K=A TO FN A(Y) STEP -1
130 X3=X2-K*K: IF X3=0 THEN 180
140 A=FN A(X3): IF A>K THEN A=K
150 FOR L=A TO 1 STEP -1
160 IF X3<>L*L THEN NEXT: NEXT: NEXT: END
170 PRINT I;"&";J;"&";K;"&";L: NEXT: NEXT: NEXT: NEXT: NEXT: END
180 PRINT I;"&";J;"&";K: NEXT: NEXT: NEXT: NEXT: END
190 PRINT I;"&";J: NEXT: NEXT: NEXT: END
200 PRINT I: NEXT: NEXT: END

```

RUN

```

### 10 ###
3 2 1 1
2 2 1 1
3 1 1 1
### 1 12 1
3 1 1 1
2 2 2 2
### 13 ###
3 2 2 2
2 2 2 2
### 14 1
3 2 1 1
### 15 1
3 2 1 1
### 16 ###
4 2 2 2
2 2 2 2
### 17 ###
4 1 2 2
3 2 1 1
4 3 2 2
3 2 2 1
### 19
4 1 1 1
BREAK IN 170
OK
>

```

```

10 CLS: CLEAR 500: PRINT TAB(10);"## POT ##": PRINT
20 INPUT "F(X) =";A$: L=LEN(A$): B=1
30 FOR I=L TO 1 STEP -1
40 IF MID$(A$,I,1)<>"^" THEN NEXT : GOTO 180
50 A=I: N=VAL(MID$(A$,A+1,1))
60 IF MID$(A$,A-1,1)=")" THEN I=0: NEXT: GOTO 150
70 FOR J=A-1 TO 1 STEP -1: B$= MID$(A$,J,1)
80 IF B$="+" OR B$="*" OR B$="-" OR B$="/" THEN B=J+1: J=1
90 NEXT: I=1: NEXT
100 C$=MID$(A$,B,A-B)+"*": D$=LEFT$(A$,B-1)
110 FOR J=1 TO N: D$=D$+C$: NEXT: D=LEN(D$)
120 D$=LEFT$(D$,D-1)+RIGHT$(A$,L-A-1)
130 PRINT D$: PRINT: GOTO 20
140 REM --- mit Klammer -----
150 FOR J=A-2 TO 1 STEP -1
160 IF MID$(A$,J,1)="(" THEN B=J: J=1: NEXT: GOTO 100
170 NEXT: PRINT"Es fehlt (": GOTO 20
180 PRINT"kein ^gefunden": GOTO 20

```

##ABLEI## Maschinencode

```

%DISPLAY 400 487
0400 00 62 04 00 00 8E 3A 3A
0408 42 02 E5 D5 C5 F5 21 62
0410 03 E5 CD DA C4 21 47 04
0418 3E A7 BE 23 20 FC 23 3A
0420 08 04 BE 23 20 F2 23 23
0428 23 23 EB E1 AF BE 28 06
0430 7E 12 23 13 18 F6 3E 3A
0438 12 13 3E 8E 12 F1 C1 D1
0440 E1 C9 0C 0A 43 6F 70 79
0448 72 69 67 68 74 20 48 2E
0450 56 6F 65 6C 7A 20 44 65
0458 7A 2E 31 39 38 37 0D 0A
0460 0A 00 89 04 0A 00 A1 20
0468 33 30 30 30 3A 20 99 3A
0470 20 9E 20 A5 31 35 29 38
0478 22 23 23 20 41 42 4C 45
0480 49 20 23 23 22 3A 20 9E
*
```

Programm ABLEI siehe übernächste Seite

```

10 CLS: PRINT TAB(10);"## DIFER ##": PRINT: GOTO 540
20 REM ----- Zerlegung -----
30 Q$="": H$="": P=0: K=0: PR$="4": L=LEN(F$)
40 FOR J=1 TO L: IF MID$(F$,J,1)="X" THEN J=L: NEXT: GOTO 60
50 NEXT: G$="0": RETURN
60 IF F$="X" THEN G$="1": RETURN
70 REM----- Klammerenebene -----
80 FOR I=1 TO L: C$=MID$(F$,I,1)
90 IF C$="(" THEN K=K+1: NEXT: GOTO 140
100 IF C$=")" THEN K=K-1: NEXT: GOTO 140
110 IF K=0 THEN GOSUB 200
120 NEXT
130 REM ----- Termtrennung -----
140 IF P=0 THEN F$=MID$(F$,2,L-2): L=LEN(F$): GOTO 30
150 L$=LEFT$(F$,P-1): R$=RIGHT$(F$,L-P)
160 IF Q$="^" THEN 500
170 IF Q$=" " THEN 380
180 GOTO 280
190 REM ----- Prioritaeten -----
200 P$="4": IF C$="^" THEN P$="3"
210 IF C$="*" OR C$="/" THEN P$="2"
220 IF C$="+" OR C$="-" THEN P$="1"
230 IF C$="=" THEN P$="0"
240 IF P$+PR$="44" THEN H$=H$+C$: P=I : RETURN
250 IF P$>PR$ THEN RETURN
260 Q$=C$: P=I: PR$=P$: RETURN
270 REM ----- Grundrechnungen -----
280 S=S+1: S$(S)=Q$: S=S+1: S$(S)=L$: S=S+1: S$(S)=R$
290 F$=L$: GOSUB 30: F$=S$(S): S=S+1: S$(S)=G$
300 GOSUB 30: GR$=G$: GL$=S$(S): S=S-1
310 R$=S$(S): S=S-1: L$=S$(S): S=S-1
320 Q$=S$(S): S=S-1
330 IF Q$="+" OR Q$="-" OR Q$=" " THEN G$=GL$+Q$+GR$
340 IF Q$="*" THEN G$="(" + GL$ + ")" * "+" + R$ + "+" + L$ + "+" * "(" + GR$ + ")"
350 IF Q$="/" THEN G$="(" + GL$ + ")" * "+" + R$ + "-" + L$ + "+" * "(" + GR$ + ")") / "(" + R$ + "+" ^ 2)"
360 RETURN
370 REM ----- Ableitungen -----
380 S=S+1: S$(S)=H$: S=S+1: S$(S)=R$: F$=R$: GOSUB 30
390 R$=S$(S): S=S-1: H$=S$(S): S=S-1: GR$="": GL$=" "
400 IF H$="SQR" THEN GL$="/2/SQR"
410 IF H$="SIN" THEN GL$="*COS"
420 IF H$="COS" THEN GL$="*-SIN"
430 IF H$="TAN" THEN GL$=""/COS": GR$="2"
440 IF H$="ATN" THEN GL$=""/(1+": GR$="2"
450 IF H$="LN" THEN GL$="/"
460 IF H$="EXP" THEN GL$="*EXP"
470 IF GL$=" " THEN GL$="*"+H$+" "
480 G$="(" + G$ + ")" + GL$ + R$ + GR$: RETURN
490 REM ----- AB -----
500 S=S+1: S$(S)=F$
510 F$="LN(" + L$ + ")" * "+" + R$: GOSUB 30
520 F$=S$(S): S=S-1: G$=F$ + "+" + G$ + "+" : RETURN
530 REM ----- Main -----
540 CLEAR 5000: DIM S$(100): S=-1
550 INPUT "F(X) =": F$: GOSUB 30
560 PRINT "F'(X) =": G$: PRINT: GOTO 550

```

Copyright H.Voelz Dez.1987.

0 REM ----- Maschinencode -----

```
10 CLEAR 3000: CLS: PRINT TAB(15); "## ABLEI ##": PRINT
20 DEF FN A(X)=X::::::::::::::::::::::::::::::::::::::::::::::::::
30 DEF FN B(X)=X::::::::::::::::::::::::::::::::::::::::::::::::::
40 M=30: DIM F$(M), F$$$(M), FV$(M), F1$(M), O(M)
50 Z$="*/": READ FZ: DIM X$(FZ), X$$$(FZ), L(FZ)
60 FOR I=1 TO FZ: READ X$(I), X$$$(I): L(I)=LEN(X$(I)): NEXT: GOTO 1190
70 Z=0: POKE 1032,65: INPUT "F(X) = "; F$(0): CALL*40A: GOSUB 110
80 PRINT "F'(X) = "; F$$$(0)
90 FOR I=1 TO LEN(F$$$(0)): POKE 865+I,ASC(MID$(F$$$(0),I,1)): NEXT
100 POKE 865+I,0: POKE 1032,66: CALL*40A: PRINT: RETURN
110 F$=F$(Z): S$=F$: GOSUB 1010
120 IF P=1 OR P=L GOTO 990
130 IF P=0 GOTO 760
140 REM===== u und v =====
150 O(Z)=0: FU$(Z)=LEFT$(F$,P-1): FV$(Z)=MID$(F$,P+1)
160 Z=Z+1: F$(Z)=FU$(Z-1): GOSUB 110: F1$(Z-1)=F$(Z)
170 F$(Z)=FV$(Z-1): GOSUB 110: Z=Z-1
180 FU$=FU$(Z): FV$=FV$(Z): F1$=F1$(Z): F2$=F$(Z+1): O=O(Z)
190 ON O GOTO 210, 210, 260, 260, 450
200 REM===== f=u+v usw. =====
210 IF F1$<>"0" AND F2$="0" THEN F$$$(Z)=F1$+MID$(Z$,0,1)+F2$: RETURN
220 IF F1$<>"0" THEN F$$$(Z)=F1$: RETURN
230 IF F2$<>"0" THEN F$$$(Z)=MID$(Z$,2,0-1)+F2$: RETURN
240 F$(Z)="0": RETURN
250 REM===== f=u*v usw. =====
260 IF F1$="0" THEN F$$$(Z)=F1$: GOTO 310
270 T=1: IF F1$="1" THEN F$$$(Z)=FV$: GOTO 310
280 S=F1$: GOSUB 1010: IF O<3 THEN F1$="("+F1$+)"
290 FF$=FV$: S$=FF$: GOSUB 1010: IF O<3 THEN FF$="("+FF$+)"
300 F$$$(Z)=F1$+"*"+FF$: GOTO 310
310 IF F2$="0" GOTO 390
320 FF$=F1$: S$=F1$: GOSUB 1010: IF O<3 THEN FF$="("+FF$+)"
330 T=T+1
340 F$$$(Z)=MID$(F$$$(Z-1),1-(T=1))+MID$(Z$,0(Z)-2,1+((T=1) AND O(Z)=3))
350 IF F2$="1" THEN F$$$(Z)=F$$$(Z)+FU$: GOTO 390
360 S$=FU$: GOSUB 1010: IF O<3 THEN FU$="("+FU$+)"
370 S$=F2$: GOSUB 1010: IF O<3 THEN F2$="("+F2$+)"
380 F$$$(Z)=F$$$(Z)+FV$: GOTO 390
390 IF F2$="0" OR O(Z)=3 GOTO 430
400 IF T=2 THEN F$$$(Z)=F$$$(Z)+FV$: GOTO 390
410 S$=FV$: GOSUB 1010: IF O<3 THEN FV$="("+FV$+)"
420 F$$$(Z)=F$$$(Z)+FV$: GOTO 390
430 F$$$(Z)=F$$$(Z): RETURN
440 REM===== f=u\ v usw. =====
450 IF F2$<>"0" GOTO 650
460 IF F1$="0" THEN F$$$(Z)=F1$: RETURN
470 IF F1$="1" THEN F$$$(Z)="" : GOTO 500
480 S$=F1$: GOSUB 1010: IF O<3 THEN F1$="("+F1$+)"
490 F$$$(Z)=F1$+"*"+F2$
500 IF FV$<>"1" THEN F$$$(Z)=FV$+"*"+F2$
510 IF LEFT$(FV$,1)="/" THEN FV$=MID$(FV$,2)
520 F2$=STR$(VAL(FV$)): IF F2$="0" GOTO 590
530 IF LEFT$(F2$,1)="/" THEN F2$=MID$(F2$,2): GOTO 530
540 IF F2$=FV$ THEN F2$=STR$(VAL(F2$)-1)
550 IF F2$<>FV$ THEN F2$=FV$
560 IF LEFT$(F2$,1)="/" THEN F2$=MID$(F2$,2): GOTO 560
570 IF LEFT$(F2$,1)="/" THEN F2$=""
580 IF LEFT$(F2$,1)="/" THEN F2$=""
590 F2$=FV$: IF RIGHT$(F2$,1)="/" THEN F2$=LEFT$(F2$,LEN(F2$)-1)
```

```

600 F2$=F2$+"-1"
610 F2$="( "+F2$+" )"
620 IF F2$="1" THEN FS$(Z)=FSS$+FU$: RETURN
630 IF F2$="0" THEN FS$(Z)=FSS$+"1": RETURN
640 FS$(Z)=FSS$+FU$+" "+F2$: RETURN
650 FF$=FU$
660 S$=FF$: GOSUB 1010: IF 0<3 THEN FF$="( "+FF$+" )"
670 S$=F1$: GOSUB 1010: IF 0<3 THEN F1$="( "+F1$+" )"
680 S$=FV$: GOSUB 1010: IF 0<3 THEN FV$="( "+FV$+" )"
690 S$=F2$: GOSUB 1010: IF 0<3 THEN F2$="( "+F2$+" )"
700 FSS$=F$(Z)+"*(" : IF F1$<>"0" THEN FSS$=FSS$+F1$+"/ "+FF$+"* "+FV$+" + "+"
710 FSS$=FSS$+"LN": IF LEFT$(FU$,1)<>"(" THEN FU$="( "+FU$+" )"
720 F$=F$+FU$
730 IF F2$="1" THEN FS$(Z)=FSS$+" ": RETURN
740 FS$(Z)=FSS$+" "+F2$+" ": RETURN
750 REM===== Funktionssuche =====
760 FOR I=1 TO FZ: IF LEFT$(F$,L(I))<>X$(I) THEN NEXT: GOTO 940
770 II=I: I=FZ: NEXT: IF RIGHT$(F$,1)<>")" GOTO 990
780 REM===== Berechnen von u =====
790 Q(Z)=II
800 Z=Z+1: F$(Z)=MID$(F$,L(II)+1,L-L(II)-1): GOSUB 110: Z=Z-1
810 F$=F$(Z): FU$=F$(Z+1): F1$=FSS$(Z+1)
820 IF F1$="0" THEN FS$(Z)="0": RETURN
830 S$=F1$: GOSUB 1010: IF 0<3 THEN F1$="( "+F1$+" )"
840 Q=Q(Z): IF 0>10 GOTO 900
850 REM===== f' = u' * g(u) =====
860 FSS$=X$(Q)+FU$+" ": IF F1$<>"1" THEN FS$=F1$+"*"+FSS$
870 IF 0>5 THEN FS$=FS$+"2"
880 FS$(Z)=FSS$: RETURN
890 REM===== f' = u' / g(u) =====
900 FSS$=X$(Q)+FU$+" ": FS$=F1$+" / "+FSS$
910 IF 0<>19 THEN FS$=FS$+"2"
920 FS$(Z)=FSS$: RETURN
930 REM===== Klammern =====
940 IF LEFT$(F$,1)<>"(" OR RIGHT$(F$,1)<>")" GOTO 970
950 S$=MID$(F$,2,LEN(F$)-2): GOSUB 1110: IF N THEN FS$(Z)="0": RETURN
960 F$(Z)=MID$(F$,2,LEN(F$)-2): GOTO 110
970 IF F$="X" THEN FS$(Z)="1": RETURN
980 FS$(Z)="0": RETURN
990 PRINT"? SYNTAX ERROR in":PRINT"? "F$: PRINT: GOTO 70
1000 REM===== Suche Rechenzeichen =====
1010 L=LEN(S$): P=0: KL=0
1020 FOR I=1 TO L: K$=MID$(S$,I,1): KL=KL+(K$=")": KL=KL-(K$="( ")
1030 IF KL<>0 THEN NEXT: GOTO 1080
1040 IF K$="+" OR K$="-" THEN P=I: O=1-(K$="-")
1050 IF K$="/" THEN IF 0>2 THEN P=I: O=3-(K$="/")
1060 IF K$="^" THEN IF 0>4 THEN P=I: O=5
1070 NEXT
1080 IF KL GOTO 990
1090 RETURN
1100 REM===== Term = Zahl ? =====
1110 FOR I=1 TO LEN(S$): K$=MID$(S$,I,1)
1120 IF K$="/" THEN IF K$<": THEN NEXT: N=1: RETURN
1130 IF K$="-" THEN NEXT: N=1: RETURN
1140 N=0: RETURN
1150 REM===== Funktionen =====
1160 DATA 7, SIN(, COS(, COS(, (-1)*SIN(,EXP(,EXP(,LN(,1/(
1170 DATA SQR(,5/SQR(, TAN(,1/COS(,ATN(,1/(
1180 !===== MAIN =====
1190 INPUT"1:ableiten 2:rechnen";A: ON A GOSUB 70, 1200: GOTO 11
1200 INPUT"X1 X2 N";X1,X2,N
1210 PRINT:PRINT"X1",X1,X2,N
1220 FOR X=X1 TO X2 STEP (X2-X1)/N
1230 PRINT X,FN A(X),FN B(X)
1240 NEXT:PRINT:RETURN

```

```

10 CLS: PRINT TAB(10);"## WAGLA ##": PRINT: F$="nicht zulaessig"
20 PRINT"mit 3 Glaesern soll Zielmenge": PRINT" erreicht werden"
30 INPUT"Zielmenge =";Z
40 INPUT"grosses Glas =";G: Q=G-Z: IF G<Z THEN PRINT F$: GOTO 40
50 INPUT"Kleines Glas =";K: IF K>Z THEN PRINT F$: GOTO 50
60 INPUT"letztes Glas =";L: IF L>K THEN PRINT F$: GOTO 60
70 FOR I=0 TO 10: M=K*I
80   FOR J=0 TO 10: N=L*J
90     IF Z=M+N THEN 140
100    IF Z=M-N THEN 150
110    IF Q=M+N THEN 160
120    IF Q=M-N THEN 170
130 NEXT: NEXT: PRINT"Es gibt keine einfache Loesung !": GOTO 30
140 PRINT Z="I"*"K"+"J"*"L: GOTO 30
150 PRINT Z="I"*"K"-"J"*"L: GOTO 30
160 PRINT Z="G"-"I"*"K"-"J"*"L: GOTO 30
170 PRINT Z="G"-"I"*"K"+"J"*"L: GOTO 30

```

```

10 CLS: PRINT TAB(9);"## DAMEN ##": PRINT: DIM A(30): Z=1
20 INPUT"Wieviele Damen";D: IF D<4 OR D>19 OR D>INT(D) THEN 20
30 REM ---- Rechnung -----
40 FOR X=1 TO 2 STEP 0
50   IF A(Z)=D THEN 140
60   A(Z)=A(Z)+1: IF Z=1 THEN 120
70   Y=1
80   FOR I=Y TO Z-1
90     IF A(I)=A(Z) THEN Y=I: NEXT X
100    IF ABS(A(I)-A(Z))=ABS(I-Z) THEN Y=I: NEXT X
110    NEXT
120    IF Z=D THEN 170
130    Z=Z+1: NEXT
140    A(Z)=0: Z=Z-1: IF Z=0 THEN END
150 NEXT
160 REM ---- Anzeige -----
170 Q=Q+1: PRINT Q"-te Variation": PRINT
180 FOR X=1 TO D
190   FOR Y=1 TO D
200     IF A(Y)=X THEN PRINT"D ";
210     IF A(Y)<>X THEN PRINT"* ";
220 NEXT: PRINT: NEXT: PRINT: PRINT: GOTO 40

```

DAMEN

Wieviele Damen 8 1 -te Variation

```

D * * * * *
* * * * * D
* * * * * D
* * * * * D
* * * * * D
* * * * * D
* * * * * D
* * * * *

```

2 -te Variation

```

D * * * * *
* * * * * D
* * * * * D
* * * * * D
* * * * * D
* * * * * D
* * * * * D
* * * * *

```

```

10 WINDOW: CLS: DIM A(6,6), B(100): PRINT TAB(14);"## SOLET ##":
20 REM ----- Wertberechnungen -----
30 Y=INT(K/10): X=INT(K)-10*Y: RETURN
40 !--- Zerlegen der Eingabe -----
50 M=(K-INT(K))*10: Y1=INT(M): X1=INT((M-Y1)*10+.5)
60 REM ----- Setzroutine -----
70 A(X,Y)=Z: H=X+X+10: S=Y+Y+21: IF R THEN 120
80 IF Z=0 THEN A=128: B=129: C=130: D=131
90 IF Z=1 THEN A=132: B=133: C=134: D=135
100 IF Z=2 THEN A=136: B=A: C=A: D=A
110 GOTO 150
120 IF Z=0 THEN A=193: B=137: C=136: D=200
130 IF Z=1 THEN A=178: B=179: C=177: D=176
140 IF Z=2 THEN A=255: B=A: C=A: D=A
150 PRINT AT(H,S);CHR$(C): PRINT AT(H-1,S);CHR$(A)
160 PRINT AT(H,S+1);CHR$(D): PRINT AT(H-1,S+1);CHR$(B): RETURN
170 IF Q THEN PRINT"nicht mehr moeglich": GOTO 380
180 PRINT"nicht erlaubt": GOTO 380
190 REM ---- Setzen der Parameter -----
200 PRINT: R=0: IF PEEK(-5)>129 THEN R=8: GOTO 230
210 FOR I=0 TO 71: READ A: POKE I,A: NEXT
220 VPOKE 14252,0: VPOKE 14253,0
230 FOR X=0 TO 6: PRINT AT(X+X+9,17);X:PRINT AT(7,X+X+21);X: NEXT
240 REM ----- verbieten -----
250 Z=2: FOR X=0 TO 1: FOR Y=0 TO 1: GOSUB 70: NEXT
260 FOR Y=5 TO 6: GOSUB 70: NEXT: NEXT
270 FOR X=5 TO 6: FOR Y=0 TO 1: GOSUB 70: NEXT
280 FOR Y=5 TO 6: GOSUB 70: NEXT: NEXT
290 REM ----- voll besetzen -----
300 Z=1: L=0: FOR Y=2 TO 4: FOR X=0 TO 6: GOSUB 70: NEXT: NEXT
310 FOR X=2 TO 4: FOR Y=0 TO 1: GOSUB 70: NEXT
320 FOR Y=5 TO 6: GOSUB 70: NEXT: NEXT
330 REM ----- Mitte leer -----
340 Z=0: X=3: Y=3: GOSUB 70
350 REM ----- Start -----
360 PRINT"Felder als Spalte*Zeile: Ist.Ziel,": PRINT"z.B. 31.33"
370 PRINT"0: zurueck 8: vor": WINDOW 7,31-R,0,15
380 INPUT"";K: Q=0
390 GOSUB 30: IF K<=0 THEN 520
400 IF K=8 THEN Q=1: GOTO 560
410 IF X>6 OR Y>6 THEN 170
420 IF A(X,Y)<>1 THEN 170
430 GOSUB 50: IF X1>6 OR X2>6 THEN 170
440 IF A(X1,Y1) OR A((X+X1)/2,(Y+Y1)/2)<>1 THEN 170
450 IF NOT(X=X1 OR Y=Y1) THEN 170
460 IF Y=Y1 AND ABS(X-X1)<>2 THEN 170
470 IF X=X1 AND ABS(Y-Y1)<>2 THEN 170
480 Z=0: GOSUB 70
490 IF X=X1 THEN Y=(Y+Y1)/2: GOSUB 70
500 IF Y=Y1 THEN X=(X+X1)/2: GOSUB 70
510 Z=1: X=X1: Y=Y1: GOSUB 70: B(L)=K: L=L+1: GOTO 380
520 IF L=0 THEN PRINT"Anfang": GOTO 380
530 L=L-1: K=B(L): GOSUB 30: GOSUB 50: Z=1: GOSUB 70
540 X=(X+X1)/2: Y=(Y+Y1)/2: GOSUB 70
550 Z=0: X=X1: Y=Y1: GOSUB 70: GOTO 380
560 IF B(L)=0 THEN PRINT"Ende": GOTO 380
570 K=B(L): GOTO 390
580 DATA 255,128,128,128,128,128,128,128,255,1,1,1,1,1,1,1
590 DATA 128,128,1 8,128,128,128,128,255,1,1,1,1,1,1,255,255
600 DATA 128,159,159,159,159,159,159,255,1,249,249,249,249,249
610 DATA 249,159,159,159,159,159,159,159,128,255,249,249,249,249
620 DATA 249,249,1,255,255,255,255,255,255,255,255,255

```

```

10 WINDOW 0,23,0,39: CLS: PRINT TAB(13);"## LNIMM ##": PRINT
20 DIM A(35,5), B(35), C(35),A$(1)
30 A$(0)="Rechner": A$(1)="Automat"
40 FOR I=1 TO 35: FOR K=1 TO 5: A(I,K)=128: NEXT: NEXT
50 WINDOW 3,12,0,39: S=35: Z=Z+1: T=T+1: IF T<T0 THEN 100
60 PAUSE 10: CLS: INPUT"wie viele Spiele automatisch:";T0: T=0
70 IF T0=0 THEN CLS: GOTO 320
80 INPUT"Anzeige der Lernmatrix 0/1";SM
90 CLS: PRINT A$(0);" gegen ";A$(1)
100 IF SM THEN GOSUB 410
110 WINDOW 5,12,0,39: PRINT"Spiel";Z
120 PRINT"35 31 25 19 13 7 1"
130 PRINT"+-----+-----+-----+-----+-----+"
140 PRINT CHR$(2);: N=0: V=0: C=INT(2*RND(1))
150 IF S=1 THEN A=1: GOTO 200
160 L=0: FOR I=1 TO 5: L=L+A(S,I): NEXT
170 R=INT(RND(1)*L): A=0: B=0
180 A=A+1: B=B+A(S,A): IF B<R THEN 180
190 IF S-A<0 THEN 170
200 N=N+1: B(N)=A: C(N)=S: S=S-A: IF S=0 THEN 260
210 IF V<>1 THEN 240
220 IF C=0 THEN PRINT"Ich nehme davon weg ";A: GOTO 320
230 C=0: GOTO 150
240 PRINT TAB(34-S);CHR$(82-17*C);
250 C=ABS(SGN(C)-1): GOTO 150
260 Q=.5: FOR I=N TO 1 STEP -1
270 IF A(C(I),B(I))*Q=.5 OR A(C(I),B(I))*Q=16384 THEN 290
280 A(C(I),B(I))=A(C(I),B(I))*Q
290 Q=1/Q: C(I)=0: B(I)=0
300 NEXT: PRINT: PRINT: PRINT"verloren hat ";A$(C)
310 V=0: GOTO 50
320 IF S<>1 THEN 340
330 V=0: PRINT"Sie haben verloren": GOTO 50
340 V=1: PRINT
350 PRINT"Der Stapel betraegt jetzt ";S
360 INPUT"Wieviel nehmen Sie (1-5) :";A
370 IF A<1 OR A>5 OR A<INT(A) THEN 360
380 IF NOT(S<6 AND S>1 AND S-A=0) THEN 400
390 PRINT"Das darf doch nicht wahr sein !": GOTO 350
400 C=1: GOTO 200
410 WINDOW 15,23,0,39: FOR J=1 TO 5: FOR I=35 TO 1 STEP -1
420 :Y=53+INT(A(I,J)/10): IF Y<33 THEN Y=33
430 IF Y>127 THEN Y=127
440 PRINTCHR$(Y);: NEXT: PRINT: NEXT: RETURN
450 ! Bei KC 85/1 und KC 87 Semikolon in Zeile 240 am Ende entfer-
nen

```

```

10 DIM A$(3): A$(1)="ASCII-Zeichen": A$(2)="Dez-Werte": A$(3)="HEX_W."
20 CLS: PRINT TAB(10);"## EINGABE ##": PRINT
30 PRINT: INPUT"Anzahl der gueltigen Stellen =";A: CLS
40 PRINT TAB(8);"-----"
50 PRINT TAB(8);"!   ASCII-Zeichen   !"
60 PRINT TAB(8);"!   Dez-Werte     !"
70 PRINT TAB(8);"!   HEX-Werte     !"
80 PRINT TAB(8);"-----"
90 PRINT: INPUT"Ihre Wahl =";B$: L=LEN (B$)
100 IF A<L THEN B$=LEFT$(B$,A): L=A
110 FOR I=1 TO 3
120   IF B$ = LEFT$(A$(I),L) GOTO 140
130 NEXT
140 CLS: ON I GOSUB 160, 230, 300, 370: GOTO 20
150 REM ----- ASCII-Zeichen -----
160 PRINT"  Es sind Zahlen zwischen 33 und 127"
170 PRINT"    einzugeben. Die zugehoerenden"
180 PRINT"      ASCII-Zeichen werden angezeigt."
190 PRINT" andere Zahlen fuehren zum Hauptmenu": PRINT
200 INPUT"Zahl   ";B: IF B<33 OR B>127 THEN RETURN
210 PRINT"Zeichen ";CHR$(B): GOTO 200
220 REM ----- Dez-Werte -----
230 PRINT"  Es sind ASCII-Zeichen einzugeben"
240 PRINT" dazu werden die dezimalen Zahlen"
250 PRINT" angezeigt. Mehr als ein Zeichen"
260 PRINT" fuehrt zum Hauptmenu zurueck": PRINT
270 INPUT"Zeichen ";A$: IF LEN (A$)>1 THEN RETURN
280 PRINT"dezimal ";ASC(A$): GOTO 270
290 REM ----- HEX-Werte -----
300 PRINT"  Es sind ASCII-Zeichen einzugeben"
310 PRINT" dazu werden die HEX-Werte gezeigt"
320 PRINT"mehr als 1 Zeichen fuehrt zum Hauptmenu": PRINT
330 INPUT"Zeichen ";A$: IF LEN (A$)>1 THEN RETURN
340 J=ASC(A$): I=J AND 15: J=INT(J/16)
350 PRINT"HEX-Wert ";CHR$(J+48-7*(J 9));CHR$(I+48-7*(I 9)): GOTO 330
360 REM ----- Fehler -----
370 PRINT"  **** fehlerhafte Eingabe ****": BEEP(3): PAUSE(10):RETURN

```

```

10 CLS: PRINT TAB(10);"## INKOR ##": PRINT: F$="EINGABE: ":A$="123"
20 PRINT F$;A$: FOR I=0 TO LEN(A$): PRINT CHR$(8);: NEXT
30 A=LEN(F$): B=CSRLIN(1): INPUT"";A$: LOCATE B,A: A$=""
40 FOR I=A TO 40: A$=A$+VGET$: PRINT CHR$(9);: NEXT: A=1: B=40
50 FOR I=1 TO LEN(A$): C=MID$(A$,I,1) " "
60 IF C AND I<B THEN B=I:!erstes echtes Zeichen
70 IF C AND I>A THEN A=I:!letztes echtes Zeichen
80 NEXT: A$=MID$(A$,B,A-B+1)
90 LOCATE 10,1: PRINT CHR$(2);A$:LOCATE 2,0: GOTO 20
100 ! Zeile 90 nur als Test

```

```

10 WINDOW 1,20,0,39: CLS: PRINT TAB(13);"## MENU ##": PRINT
20 DEF FN A(X)=(X AND 15) - 9*(X>64)
30 DEF FN B(X)=X+48 -7*(X>9)
40 GOTO 210
50 !----- Verteiler -----
60 WINDOW 10,12,0,39: INK 4: PAPER 3: CLS
70 ON I GOSUB 90, 140, 340: GOTO 210
80 !----- hex- dez -----
90 INPUT"HEX =";A$: X=0
100 FOR I=1 TO LEN(A$)
110   X=X*16 + FN A(ASC(MID$(A$,I,1)))
120 NEXT: PRINT"DEZ =";X: RETURN
130 !----- dez- hex -----
140 INPUT"DEZ =";N: A$="": J=N: PRINT"HEX = ";: IF J>4095 THEN GOSUB 2
150 GOSUB 170: PRINT A$: RETURN
160 I=INT(N/4096): J=N-4096*I: N=I
170 FOR X=1 TO 3
180   I= J AND 15: J=INT(J/16): A$=CHR$(FN B(I)) + A$
190 NEXT: RETURN
200 !----- Main -----
210 WINDOW 3,5,10,30
220 PRINT"#   HEX nach DEZ": I=1
230 PRINT"   DEZ nach HEX"
240 PRINT"       Ende";
250 WINDOW 17,20,0,39: INK 3: PAPER 4: CLS
260 PRINT"Eingabe: Hoch; Tief; Bestaetigen"
270 WINDOW 19,19,10,14: INK 8: PAPER 7: CLS
280 INPUT"";A$: IF A$<>"H" AND A$<>"T" AND A$<>"B" GOTO 280
290 IF A$="B" GOTO 60
300 PRINT AT(I+2,10);" "
310 IF A$="H" THEN I=I-1: IF I=0 THEN I=1
320 IF A$="T" THEN I=I+1: IF I>3 THEN I=3
330 PRINT AT(I+2,10);"#": GOTO 280
340 WINDOW: CLS: END

```

```

10 CLS: PRINT TAB(14);"## ZEICHEN ##": PRINT
20 PRINT"Erzeugung der Zeichen"
30 PRINT"bitte einen Moment warten"
40 FOR I=0 TO 7
50 FOR J=0 TO 16 STEP 8: B=0: READ A$
60 FOR X=1 TO 8: B=B+B: A=0: IF MID$(A$,X,1)="Ä" THEN A=1
70 B=B+A: NEXT: POKE I+J,B: NEXT: NEXT
80 VPOKE 14252,0: VPOKE 14253,0
90 REM --- ABLAGE DER ZEICHEN ---
100 DATA ÄÄ...ÄÄ, ...ÄÄÄ., Ä.Ä.Ä.Ä.
110 DATA .ÄÄ..ÄÄ., ..ÄÄÄ..., .Ä.Ä.Ä.Ä
120 DATA ..ÄÄÄÄ., ÄÄÄ...., Ä.Ä.Ä.Ä.
130 DATA .Ä.Ä... , ..ÄÄÄ..., .Ä.Ä.Ä.Ä
140 DATA ÄÄ...ÄÄ, ...ÄÄÄ., Ä.Ä.Ä.Ä.
150 DATA .ÄÄ..ÄÄ., ..... , .Ä.Ä.Ä.Ä
160 DATA ..ÄÄÄÄ., .ÄÄÄÄÄ., Ä.Ä.Ä.Ä.
170 DATA ...ÄÄ..., .ÄÄÄÄÄ., .Ä.Ä.Ä.Ä
180 PRINT"Darstellung der Zeichen": PRINT
190 FOR J=1 TO 3: PRINT" ";
200 FOR I=128 TO 130: PRINT" "CHR$(I);: NEXT
210 NEXT

```

Ä ≈ ■

```

10 CLS: PRINT TAB(10);"## CLEAR ##": PRINT: CLEAR
20 C=65536: B=DEEK(854): R=DEEK(944)
30 PRINT"Stringraum =";R-B;"Bytes"
40 A=R: PRINT"RAM-Ende =";A;"Dez. = ";: IF A<0 THEN A=A+C
50 D=C: FOR I=1 TO 4 : D=D/16
60 B=INT(A/D): A=A-B*D: B=B+48: IF B>57 THEN B=B+7
70 A$=A$+CHR$(B)
80 NEXT:PRINTA$;" HEX"
90 PRINT: INPUT"neuer Stringraum =";E
100 INPUT"neues RAM-Ende in HEX =";A$: A=0
110 IF LEN(A$)>4 THEN PRINT"zu gross": GOTO100
120 FOR I=1 TO LEN(A$)
130 B=ASC(MID$(A$,I,1))-48: IF B>9 THEN B=B-7
140 A=A*16+B
150 NEXT: IF A>C/2 THEN A=A-C
160 CLEAR E,A: GOTO20

```

```

10 CLS: PRINT TAB(10)"## DATEN ##": PRINT
20 A=B=C: I=0
30 B=DEEK(987): A=DEEK(854)-100-B
40 PRINT"maximal";A;"Werte moeglich"
50 PRINT:INPUT"1=Eingabe 2=Anzeige";C
60 ON C GOSUB 70, 120: GOTO 50
70 FOR I=0 TO A
80   PRINT"NR";I;: INPUT"";C
90   IF C<0 OR C>254 THEN C=255
100  POKE B+I,C: IF C=255 THEN I=A
110 NEXT: RETURN
120 FOR I=0 TO A: C=PEEK(B+I)
130   PRINT C;: IF C=255 THEN I=A
140 NEXT: PRINT: RETURN

```

```

10 CLS: PRINT TAB(15);"## LED ##"
20 C=136: A=INP(C)
30 FOR I=1 TO 20
40 OUT C, A OR 32: PAUSE 2
50 OUT C, A: PAUSE 2
60 NEXT

```



```

0001 ;=====*
0002 ; BASIC:  GOSUB A; GOTO A  *
0003 ; ##      BASGO      ##  *
0004 ; H.Voelz           Dez: 87  *
0005 ;=====*
0006 ;
0007          ORG          406H
0008 ;
C96F      0009 EIN      EQU      0C96FH
CA0A      0010 GOTO     EQU      0CA0AH ; Unterprogr.GOTO
C854      0011 RUNMOD   EQU      0C854H ; RUN-Mode
C327      0012 TEMO     EQU      0C327H ; Fehlertest
0358      0013 AKZEI    EQU      358H  ; aktuelle Zeile
008C      0014 TGS      EQU      8CH   ; Token GOSUB
          0015 ;
          0016 JUMP     POP      DE      ; Stack
          0017         POP      DE      ; korrigieren
          0018         POP      DE
          0019         CALL     EIN      ; USR- DE
          0020 JPR      LD      HL,RUNMOD
          0021         PUSH     HL      ; auf Stack
          0022         JP      GOTO
          0023 ;
          0024 GOSUB    CALL     EIN
          0025         POP      HL
          0026         POP      HL
          0027         POP      BC
          0028         IN      A,(88H) ; IMR AN
          0029         SET     2,A
          0030         OUT     (88H),A
          0031         LD      C,3
          0032         CALL     TEMO
          0033         PUSH     HL
          0034         PUSH     HL
          0035         LD      HL,(AKZEI)
          0036         EX      (SP),HL
          0037         LD      A,TGS
          0038         PUSH     AF
          0039         INC     SP
          0040         JR      JPR-$
          0041 ;

```

ERRORS=0000

AKZEI	0358	EIN	C96F	GOSUB	0413
GOTO	CA0A	JPR	040C	JUMP	0406
RUNMOD	C854	TEMO	C327	TGS	008C

```

10 CLS: PRINT TAB(12);"## KORRE ##": PRINT
20 Z=1E15: NN=100: Z2=1/Z
30 DIM W(1,NN), A(3), B(3), C(3), A$(3), B$(1): B$(0)="X": B$(1)="Y"
40 A$(0)="Y=A*X": A$(1)="Y=A/X": A$(2)="Y=A^X": A$(3)="Y=X^A"
50 REM--- START ----
60 PRINT: INPUT"1:Eingabe 2:Anzeige 3:Werte 4:X - Y";A
70 CLS: ON A GOSUB 90, 140, 200, 390: GOTO 60
80 !----- EINGABE -----
90 FOR X=N TO NN: PRINT X;: FOR Y=0 TO 1
100 A=Z: PRINT B$(Y);: INPUT"=";A: IF A=Z THEN 120
110 W(Y,X)=A: NEXT: NEXT: N=NN: RETURN
120 N=X: X=NN: Y=1: NEXT: NEXT: RETURN
130 !----- ANZEIGE -----
140 PRINT:PRINT"Nr.", "X", "Y": A=20
150 FOR I=0 TO N-1
160 PRINT I,W(0,I),W(1,I): A=A-1
170 IF A=0 THEN INPUT"";A: A=20
180 NEXT: RETURN
190 REM--- FIT -----
200 FORI=0 TO 3: A(I)=0: B(I)=0: C(I)=1: NEXT
210 FOR I=0 TO N-1
220 X=W(0,I): Y=W(1,I): IF ABS(X)<Z2 THEN X=Z2
230 IF Y<Z2 THEN C(2)=0: C(3)=0
240 IF X<=Z2 THEN C(3)=0
250 A=Y/X: A(0)=A(0)+A: B(0)=B(0)+A*A
260 A=Y*X: A(1)=A(1)+A: B(1)=B(1)+A*A
270 IF C(2) THEN A=LN(Y)/X: A(2)=A(2)+A: B(2)=B(2)+A*A
280 IF C(3) THEN A=LN(X): IF ABS(A)<Z2 THEN A=Z2
290 IF C(3) THEN A=LN(Y)/A: A(3)=A(3)+A: B(3)=B(3)+A*A
300 NEXT
310 PRINT"Funktion","Faktor","Stand.Abwchg."
320 FOR I=0 TO 3
330 A(I)=A(I)/N: B(I)=SQR((B(I)/N-A(I)*A(I))/(N+1))
340 IF C(I) THEN PRINT A$(I),A(I),B(I)
350 NEXT: IF C(2) THEN C(2)=EXP(A(2))
360 IF C(3) THEN C(3)=1/A(3)
370 RETURN
380 !----- REGRESSION -----
390 PRINT TAB(10);"Regression"
400 FOR I=0 TO 3: PRINTI+1;";";A$(I): NEXT
410 A=Z:INPUT A: IF A<1 OR A>4 THEN RETURN
420 IF C(A-1)=0 THEN PRINT "nicht zulaessig": GOTO 410
430 B=Z: INPUT"1: X- Y 2: Y- X";B: IF B<1 OR B>4 THEN 400
440 IF B=2 GOTO 510
450 X=Z: INPUT"X =";X: IF X=Z THEN 430
460 IF A=1 THEN PRINTA(0)*X
470 IF A=2 THEN PRINTA(1)/X
480 IF A=3 THEN PRINT C(2)^X
490 IF A=4 THEN PRINT X^A(3)
500 GOTO 450
510 Y=Z: INPUT"Y =";Y: IF Y=Z THEN 430
520 IF A=1 THEN PRINT Y/A(0)
530 IF A=2 THEN PRINT A(1)/Y
540 IF A=3 THEN PRINT LN(Y)/A(2)
550 IF A=4 THEN PRINT Y^C(3)
560 GOTO 510

```

```

10 CLS: PRINT TAB(12);"## STAMZA ##": PRINT
20 N=100: DIM A(N), B(3): A(0)=1: Z=1E38
30 PRINT: INPUT"1:Eingabe 2:Anzeige 3:Rechnen";A: CLS
40 ON A GOSUB 60, 110, 160: GOTO 30
50 REM---- EINGABE -----
60 PRINT"Eingabe der Daten": PRINT"Ende nur ENTER ": PRINT
70 FOR I=A(0) TO N
80 B=Z: INPUT B: IF B=Z THEN A(0)=I-1: I=N: NEXT: RETURN
90 A(I)=B: NEXT: A(0)=N: RETURN
100 REM--- ANZEIGE -----
110 PRINT"Nr.", "Wert": A=20
120 FOR I=1 TO A(0)
130 PRINT I, A(I): A=A-1: IF A THEN NEXT: RETURN
140 INPUT"";A: A=20: GOTO 130
150 REM----- RECHNEN -----
160 FOR X=0 TO 3: B(X)=0: NEXT
170 FOR I=1 TO A(0): B=1: A=A(I)
180 FOR X=0 TO 3: B=B*A: B(X)=B(X)+B
190 NEXT: NEXT
200 FOR X=0 TO 3: B(X)=B(X)/A(0): NEXT
210 A=B(0): B=A*A: C=B*B
220 B(3)=B(3)-4*A*B(2)+6*B(1)*B-3*C
230 B(2)=B(2)-3*A*B(1)+2*A*B
240 B(1)=B(1)-B
250 PRINT"Name", "Wert", "Moment-Nr.": PRINT
260 PRINT"Mittelwert", B(0), "1"
270 PRINT"Streuung", B(1), "2"
280 PRINT"Stand. Abwchg.", SQR(B(1)*A(0)/(A(0)-1))
290 PRINT,B(2), "3": PRINT,B(3), "4"
300 PRINT"Schiefe", B(2)/B(1)/SQR(B(1))
310 PRINT"Steilheit", B(3)/B(1)/B(1)
320 PRINT"Kurtosis": PRINT: RETURN

```

```

10 CLS: PRINT TAB(10); "## FIT ##": NN=50: Z=1E38: Z2=1E-30: GOTO 530
20 X=LN(X)
30 Y=LN(Y)
40 P=P+X: Q=Q+Y: R=R+X*X: S=S+Y*Y: T=T+X*Y: RETURN
50 X=1/X: GOTO 40
60 X=LN(X): GOTO 40
70 X=1/X
80 Y=C(J)+B(J)*X: RETURN
90 Y=C(J)+B(J)*LN(X): RETURN
100 Y=C(J)*B(J)^X: RETURN
110 Y=C(J)*X^B(J): RETURN
120 X=(Y-C(J))/B(J): RETURN
130 X=B(J)/(Y-C(J)): RETURN
140 X=EXP((Y-C(J))/B(J)): RETURN
150 X=LN(Y/C(J))/LN(B(J)): RETURN
160 X=(Y/C(J))^(1/B(J)): RETURN
170 !===== FIT =====
180 PRINT: PRINT "Zwei-Parameter-Regression"
190 PRINT: PRINT "Es existieren folgende Funktionen"
200 INPUT "1:a+bx 2:a+b/x 3:a+blnx 4:ab^x 5:ax^b"; F
210 P=0: R=0: S=0: T=0: Q=0: FOR I=0 TO N-1: X=A(0,I): Y=A(1,I)
220 ON F GOSUB 40, 50, 60, 30, 20: NEXT
230 R=N*R-P*P: S=N*S-Q*Q: T=N*T-P*Q: P=P/N: Q=Q/N: B(0)=T/R: B(1)=S/T
240 IF ABS(T)<Z2 THEN T=Z2
250 S=(S-R)/(2*T): B(2)=S+SGN(B(0))*SQR(S*S+1)
260 FOR I=0 TO 2: C(I)=Q-P*B(I)
270 NEXT I: IF F=4 OR F=5 THEN FOR I=0 TO 2: C(I)=EXP(C(I)): NEXT
280 IF F=4 THEN FOR I=0 TO 2: B(I)=EXP(B(I)): NEXT
290 PRINT: PRINT "Korrelation R^2 =" B(0)/B(1): PRINT
300 PRINT "Es gibt 3 Schwankungen um"
310 PRINT "1:x", "2:y", "3:x und y"
320 PRINT: PRINT " a=": FOR I=0 TO 2: PRINT C(I),: NEXT: PRINT
330 PRINT " b=": FOR I=0 TO 2: PRINT B(I),: NEXT: PRINT
340 !==== REGRESSION =====
350 INPUT "Welchen Fall zur Regression nutzen"; J: J=J-1
360 G=3: INPUT "1: x- y 2: y- x 3: Abbruch"; G
370 ON G GOTO 380, 400, 180
380 X=Z: INPUT " X="; X: IF X=Z GOTO 360
390 ON F GOSUB 80, 70, 90, 100, 110: PRINT "y=" Y: GOTO 380
400 Y=Z: INPUT " Y="; Y: IF Y=Z GOTO 360
410 ON F GOSUB 120, 130, 140, 150, 160: PRINT " X="; X: GOTO 400
420 !===== EINGABE =====
430 FOR X=N TO NN: PRINT X;: FOR Y=0 TO 1
440 A=Z: PRINT A$(Y);: INPUT "="; A: IF A=Z THEN 460
450 A(Y,X)=A: NEXT: NEXT: N=NN: RETURN
460 N=X: X=NN: Y=1: NEXT: NEXT: RETURN
470 !==== ANZEIGE =====
480 A=20: PRINT "Nr.", "X", "Y"
490 FOR X=0 TO N-1: PRINT X, A(0,X), A(1,X): A=A-1
500 IF A=0 THEN INPUT "A="; A: A=20
510 NEXT: RETURN
520 !===== MAIN =====
530 DIM A(1,NN), B(2), C(2), A$(1): A$(0)="X": A$(1)=" Y"
540 PRINT: INPUT "1:Eingeben 2:Anzeigen 3: Fitten"; A
550 CLS: ON A GOSUB 430, 480, 180: GOTO 540

```

BASIC für Fortgeschrittene

=====

Name	Send.	Rechner	Inhalt	Kassette
------	-------	---------	--------	----------

ACK1	5	*	Ackemann-Funktion (rekursiv)	
ACK2	5	*	dito mit Stack	
BASUHR	4	M	BASIC mit Uhr und Zeitmessung	
BASUR	4	R	BASIC-Uhr KC 87	
BINOM	5	*	Binomial-Koeffizient	
BUB2	1	*	Bubble mit Pointern, verkürzt	
COPY	4	M	Bildschirmcopy !!LOAD!!	
DEZPER	5	*	Periodische Dezimalbrüche	
EULER	2	*	lange Eulersche Zahl	
FAKUL	2	*	Fakultät nach Stirling	
FIBON	5	*	Fibonacci-Zahlen	
FILE	1	*	erzeugt Zufallszahlen,-texte	
FNSA	4	*	bis 26 Fkt. direkt eingeben	
FRAGEB	3	M	Fraktale als Gebirge + normal	
FRAK3	3	M	Fraktale: Feldern von FRAK87	
FRAK87	3	R	Felder für Fraktale KC 87	
FRAKT	3	M	Fraktale s/w hoher Auflösung	
FRAKTAL	3	M	farbige Fraktale	
FUNKTION	4	*	direkte Fkt.-Eingabe + Tabelle	
GGT	5	*	Größter gemeinsamer Teiler	•
KOMBA	2	*	alle kombinatorische Formeln	
PI	2	*	viele Ziffern von PI	
PLOT3D	4	M	3D-Perspektive von Fkt.	
PUFOR	2	*	Formatieren auf den Punkt	
QUICK	1	*	Quicksort (binär)	
RASTER	4	M	Raster-Hintergrund m.Farbbyte	
REFOR	2	*	rechtsseitiges Formatieren	
RUND	2	*	Runden von Zahlen	
SHAKER	1	*	Shakersort (hin und her)	
TEFRA	3	*	testen für Fraktale	
WOBUB	1	*	Wortbubble	
ZABUB	1	*	Zahlenbubble	

Programme der Sendungen 6 - 10 :

BASIC für Fortgeschrittene Teil 2

=====

Name	Send.	Rechner	Inhalt
------	-------	---------	--------

ABLEI	6	*	Konfortable Ableitung
BASGO	9	*	Zeilensprung über Variable
CLEAR	9	*	Test der CLEAR-Anweisung
DAMEN	7	*	Damen auf Schachbrett
DATEN	9	*	Viele Daten mit POKE eingeben
DIFER	6	*	einfache formale Abbleitung
EINGABE	8	*	Eingabe mit Länge gültiger Zeichen
FIT	10	*	Zweiparameter-Anpassung
INKOR	8	M	Eingabe mit Korrektur
KORRE	10	*	Korrelation bei zwei Werten
LED	9	M	Steuerung der Tape-Diode
LNIMM	7	*	Lernendes Nimm
MENU	8	*	Eingabe über Zeile
POTAN	6	*	formale Umrechnung bei Potenzen
PRIMEL	6	*	mehrfache Primzahlprodukte
SOLET	7	*	Soletude=Solohalma
SPRICH	9	*	Fleddern von Sprichwörtern
STAMZA	10	*	Statistische Maßzahlen
VIQUA	6	*	Vier Quadratzahlen
WAGLA	7	*	Wasserglasproblem
ZEICHEN	8	M	Definition von Zeichen

* Für KC 85/1 bis 3 und KC 87

M Nur für KC 85/2 bis 3