

Aufgabensammlung zur

Einführung in die Rechnerpraxis

"Mathematische Probleme in BASIC lösen"



Andreas Carl

Klemens Kieninger

Holger Kett

Überarbeitet und erweitert von
Ingo Ende

PDF-Umsetzung von
Thomas Antoni - thomas@antonis.de <http://www.antonis.de>

Inhaltsverzeichnis

1. Berechnung von Formeln

- 1.1 Nullstellenberechnung
- 1.2 Polynomberechnung
- 1.3 Berechnung einer quadratischen Gleichung
- 1.4 Berechnung von Mittelwerten
- 1.5 Größter gemeinsamer Teiler

2. Wertetabellen berechnen und ausgeben

- 2.1 Multiplikationstabelle
- 2.2 Sinustabelle

3. Grenzwerte von Folgen, berechnen von Reihenwerten

- 3.1 Grenzwert einer gegebenen Folge
- 3.2 Reihenentwicklung
- 3.3 Berechnung von Reihenwerten
- 3.4 Rechnen mit einer Zahlenreihe

4. Iterationsverfahren zur Nullstellenbestimmung von Funktionen

- 4.1 Intervallhalbierungsmethode
- 4.2 Sekantenmethode
- 4.3 Newtonsches Iterationsverfahren

5. Eindimensionale Felder

- 5.1 Primzahlen bestimmen (Methode nach Erathostenes)
- 5.2 Sortieren durch direkten Vergleich (Bubblesort)
- 5.3 Sortieren durch benachbarten Vergleich
- 5.4 Sortieren durch Einfügen
- 5.5 Sortieren von Strings (am Beispiel 5.4)

6. Rechnen mit Vektoren

- 6.1 Addition von Vektoren - Winkel zwischen Vektoren
- 6.2 Vektorprodukt
- 6.3 Skalarprodukt

7. Mehrdimensionale Felder

- 7.1 Addition zweier Matrizen
- 7.2 Matrizenaddition mit beliebig vielen Matrizen
- 7.3 Skalarprodukt berechnen
- 7.4 Multiplikation zweier Matrizen
- 7.5 Determinante nach Sarrus bestimmen

8. Differentiation von ganzrationalen Funktionen

9. Numerische Integration von beliebigen Funktionen

- 9.1 Trapezformel
- 9.2 Simpsonformel

10. Umrechnung von Zahlensystemen

- 10.1 Umwandlung Dualzahl in Dezimalzahl
- 10.2 Dezimalzahl in Dualzahl
- 10.3 Dezimalzahl in Hexadezimalzahl
- 10.4 Hexadezimalzahl in Dezimalzahl

1 Berechnung von Formeln

1.1 Nullstellenberechnung

Aufgabe:

Die Formel: $f(x) = x^2 - 6x + 5$

besitzt ein Minimum an der Stelle (3,-4) und die Nullstellen $x=1$ und $x=5$. Gesucht sind diejenigen aufeinanderfolgenden Werte von x , zwischen denen der Wert der Funktion das Vorzeichen von positiv nach negativ oder von negativ nach positiv wechselt, wenn x die ganzzahligen Werte von -4 bis 10 annimmt

Hinweis:

Definiere die Funktion mit dem Befehl DEF FNA(x)= x^2-6x+5 und verwende die SGN - Funktion zum Erkennen eines Vorzeichenwechsels. Bsp.: SGN(y) überprüft das Vorzeichen der Zahl in y .

Programmlisting:

```
REM Berechnung von Formeln: ➔ Programm 1.1
CLS
DEF FNA (x) = x ^ 2 - 6 * x + 5
PRINT
PRINT ' Fügt Leerzeilen ein
FOR x1 = -4 TO 10 STEP 2 'Zählt von -4 bis 10 im Abstand von 2
    x2 = x1 + 2
    y1 = FNA(x1)
    y2 = FNA(x2)
    IF SGN(y1) * SGN(y2) = -1 THEN
        PRINT "Die Funktion f(x) = x^2 - 6x + 5 wechselt das Vorzeichen zwischen"; x1; "und"; x2
    END IF
NEXT x1
END
```

1.2 Polynomberechnung

Aufgabe:

Gesucht ist ein Programm, daß mit vorgegebenen Nullstellen das zugehörige Polynom berechnet. Es genügt, wenn man nur für reelle Zahlen das Polynom berechnen kann.

Bsp.: Nullstellen $x_1 = 1$
 $x_2 = 2$
 $x_3 = 3$
 $x_4 = 4$
 $x_5 = 5$
 $x_6 = 6$

Polynom

$$f(x) = x^6 - 21x^5 + 175x^4 - 735x^3 + 1624x^2 - 1764x + 720$$

Verlangt ist eine Abfrage nach der Anzahl der Nullstellen. Danach soll dann eine Abfrage der 1.Nullstelle, der 2.Nullstelle, usw. erfolgen. Anschließend werden die Koeffizienten des Polynoms in einer übersichtlichen Form ausgegeben.

Programmlisting:

```
DO
REM Berechnung von Formeln: → Programm 1.2
CLS
PRINT "*****"
PRINT "* Polynomberechnung * "
PRINT "*****"
PRINT
INPUT "Wieviel Nullstellen sind vorgegeben "; n
REDIM a(n), x(n) 'Redimensioniert bzw. setzt Felder auf Null zurück
PRINT
FOR i = 1 TO n 'Abfrage der Nullstellen
PRINT STR$(i) + ". Nullstelle eingeben!:"
INPUT "NST= ", x(i)
a(i) = 0
NEXT i
a(0) = 0
a(n) = 1
FOR i = n TO 1 STEP -1 'zählt in Einerschritten rückwärts
FOR j = n TO 1 STEP -1
a(n - j) = a(n - j) - a(n - j + 1) * x(n - i + 1)
NEXT j
NEXT i
PRINT 'Leerzeile
CLS
PRINT "*****"
PRINT "* Polynomberechnung * "
PRINT "*****"
PRINT
PRINT "Koeffizienten des Polynoms:"
PRINT
FOR i = n TO 1 STEP -1
PRINT "a("; i; ")="; a(i)
NEXT i
PRINT
PRINT
INPUT "Weitere Aufgabe rechnen? j/n :", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

1.3 Berechnung einer quadratischen Gleichung

Aufgabe:

Erstelle ein Programm zur Lösung einer quadratischen Gleichung der allgemeinen Form:

$$ax^2+bx+c=0$$

Problem:

Wie erhält man eine übersichtliche Lösungsmenge der Gleichung, bei allen möglichen Einsetzungen für die Koeffizienten a, b und c ?

Hinweis:

Beachte, es muß eine Fallunterscheidung programmiert werden.

1. (a, b, c) = 0
2. (a, b) = 0
3. (a) = 0
4. Diskriminante > 0
5. Diskriminante = 0
6. Diskriminante < 0 (es entsteht → eine komplexe Lösung)

Programmlisting:

REM Berechnung von Formeln: → Programm 1.3

```
DECLARE SUB einlesen () ' ← SUB wird in QBASIC automatisch erzeugt
DECLARE SUB berechnen ()
' Lösung einer quadratischen Gleichung mit reellen Koeffizienten
DEFSNG DIS ' → Diskriminante DIS wird als SINGL-Datentyp definiert
' DEFSNG A - D ' → Alternative zu COMMON SHARED
DEFSNG X
DEFSTR Z ' → Z ist eine Stringvariable
COMMON SHARED a, b, c AS SINGLE ' → Koeffizienten a,b,c
' → COMMON SHARED macht Variablen allen SUB's zugänglich
DO
  CLS ' → Bildschirm löschen
  PRINT
  "*****"
  PRINT "* Lösung einer quadratischen Gleichung der Form ax^2+bx+c *"
  PRINT
  "*****"
  PRINT
  einlesen ' → Koeffizienten einlesen
  berechnen ' → Lösung(en) berechnen
  LOCATE 20, 5 ' → Gehe zu Zeile 10 Spalte 5 (Alternative zu Print)
  INPUT "Weitere Aufgabe rechnen? (j/n): ", zeichen
LOOP UNTIL zeichen = "n" OR zeichen = "N"
END

DEFSNG Z
SUB berechnen
'SHARED a, b, c ← Alternative zu COMMON SHARED (Übergabe der Variablen)
DIS = b ^ 2 - 4 * a * c
IF a = 0 THEN
  IF b <> 0 THEN
    x = -c / b
    PRINT " Es gibt eine reelle Lösung."
    PRINT
    PRINT " Die Lösung lautet: x = "; x
    PRINT
  ELSE
    PRINT " Alle Zahlen sind Lösungen, wenn";
    PRINT " c gleich Null. Andernfalls gibt es";
    PRINT " keine Lösungen."
    PRINT
  END IF
END IF
EXIT SUB ' → Unterprogramm verlassen
END IF
```

```
IF DIS > 0 THEN
  DIS = SQR(DIS)
  x1 = (-b + DIS) / (2 * a)
  x2 = (-b - DIS) / (2 * a)
  PRINT " Es gibt zwei reelle Lösungen."
  PRINT
  PRINT " Die Lösungen lauten: x1 = "; x1
  PRINT " und: x2 = "; x2
  PRINT
END IF
IF DIS < 0 THEN
  DIS = SQR(-DIS)
  x1r = -b / (2 * a)
  x2r = x1r
  x1i = DIS / (2 * a)
  x2i = -DIS / (2 * a)
  PRINT " Es gibt zwei komplexe Lösungen."
  PRINT
  PRINT " Die Lösungen lauten: x1 = "; x1r; "+ j("; x1i; ")"
  PRINT " und: x2 = "; x2r; "+ j("; x2i; ")"
  PRINT
END IF
IF DIS = 0 THEN
  x1 = -b / (2 * a)
  PRINT " Es gibt eine reelle →."
  PRINT
  PRINT " Die Lösung lautet: x = "; x1
  PRINT
END IF
END SUB

SUB einlesen
'SHARED a, b, c ← Alternative zu COMMON SHARED
(Übergabe der Variablen)
INPUT " Geben Sie die Koeffizienten a,b,c ein: ", a, b, c
PRINT " Die quadratische Gleichung lautet also: ";
PRINT STR$(a) + "x^2 + " + STR$(b) + "x + " + STR$(c) + " = 0"
PRINT
END SUB
```

1.4 Berechnung von Mittelwerten

Aufgabe:

Erstelle anhand der folgenden Formeln ein Programm, daß die verschiedenen Mittelwerte einer bestimmten Zahlenreihe ermittelt.

- a.) Arithmetisches Mittel:
$$A = \frac{1}{n} (x_1 + x_2 + x_3 + \dots + x_n)$$
- b.) Geometrisches Mittel:
$$G = \sqrt[n]{x_1 \cdot x_2 \cdot x_3 \cdot \dots \cdot x_n}$$
- c.) Harmonisches Mittel:
$$\frac{1}{H} = n \left(\frac{1}{x_1} + \frac{1}{x_2} + \frac{1}{x_3} + \dots + \frac{1}{x_n} \right)$$
- d.) Quadratisches Mittel:
$$Q = \sqrt{\frac{1}{n} (x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2)}$$

Hinweis:

Die Berechnung des harmonischen Mittels muß abgebrochen werden, wenn eine Zahl mit dem Wert Null eingegeben wurde. Die Eingabe der Zahl Null muß allerdings gestattet sein

Programmlisting:

REM Berechnung von Formeln: → Programm 1.4

' Bestimmung von Mittelwerten beliebiger Zahlen
DECLARE SUB mittelwerte () '→ SUB deklarieren
DO

```
    REDIM SHARED zahl(3000) AS DOUBLE
    CLS
    PRINT "*****"
    PRINT "* Programm zur Bestimmung von Mittelwerten beliebiger Zahlen *"
    PRINT "*****"
    PRINT
    PRINT "Mit diesem Programm können für bis zu 3000 Zahlen Mittelwerte bestimmt werden."
    INPUT "Wieviel Zahlen wollen Sie eingeben? :", menge
    PRINT "Geben Sie nun Ihre Zahlen ein."
    FOR i = 1 TO menge
        PRINT "Geben sie die " + STR$(i) + ". von " + STR$(menge) + " Zahl(en) ein "
        INPUT "Zahl = :", zahl(i)
    NEXT i
    PRINT
    PRINT "Für die " + STR$(menge) + " Zahlen w→→ werden nun die Mittelwerte erstellt"
    SLEEP 2 '→ zwei Sekunden Pause einlegen
    mittelwerte '→ SUB aufrufen
    LOCATE 20, 5 '→ Positioniert nächste Eingabe auf Zeile 20 und Spalte 5
    COLOR 20 '→ Nachfolgende Anweisung erfolgt rot + blinkend, (4 + 16)
    INPUT "Nochmals durchrechnen? j/n: ", frage$
    COLOR 7, 0 '→ Zurücksetzen auf Hellgrau (7,Vordergr.) und Schwarz (Hinter.)
LOOP UNTIL frage$ = CHR$(78) OR frage$ = CHR$(110)
END
```

```
SUB mittelwerte
    SHARED menge '→ übergibt den Wert von "menge" an diese SUB
    FOR i = 1 TO menge
        gesamt = gesamt + zahl(i) 'für arithmetischen Mittelwert
        gesamtquadr = gesamtquadr + zahl(i) ^ 2 'für quadratischen Mittelwert
        IFF zahl(i) <> 0 then
            gesamtharmo = gesamtharmo + 1 / zahl(i) 'für harmonischen Mittelwert
        End If
    NEXT i
    CLS
    PRINT "*****"
    PRINT "* Programm zur Bestimmung von Mittelwerten beliebiger Zahlen *"
    PRINT "*****"
    PRINT
    PRINT
    PRINT "Das arithmetische Mittel der " + STR$(mitzaehlen) + " Zahlen ist xquer= ", gesamt / menge: PRINT
    PRINT "Das geometrische Mittel der " + STR$(mitzaehlen) + " Zahlen ist xgeom= ", gesamt ^ (1 / menge): PRINT
    PRINT "Das quadratische Mittel der " + STR$(mitzaehlen) + " Zahlen ist xquadr= ", gesamtquadr / menge: PRINT
    PRINT "Das harmonische Mittel der " + STR$(mitzaehlen) + " Zahlen ist xharmo= ", 1 / ((1 / menge) * gesamtharmo): PRINT
    SLEEP 2 '→ zwei Sekunden Pause einlegen
END SUB
```

1.5 Größter Gemeinsamer Teiler

Problem:

Gegeben seien zwei ganze Zahlen a und b. Gesucht ist nun der größte gemeinsame Teiler beider Zahlen. Zum Beispiel, der größte gemeinsame Teiler von 24 und 56 ist 8.

Aufgabe:

Schreibe nun ein Programm, daß den größten gemeinsamen Teiler sucht und diesen dann ausgibt.

Programmlisting:

REM Berechnung von Formeln: → Programm 1.5

```
DO
  CLS
  PRINT "*****"
  PRINT "* Berechnung größter gemeinsamer Teiler *"
  PRINT "*****"
  PRINT
  INPUT "Geben Sie die Werte für a und b ein! (Durch Komma getrennt) :", a, b
  PRINT
  FOR i = 1 TO a
    d = INT(a / i) '→ überprüft ob a durch i teilbar ist
    e = a - i * d
    f = INT(b / i) '→ überprüft ob b durch i teilbar ist
    g = b - i * f
    IF e <> 0 OR g <> 0 THEN
      '→ Schleife muß weiter durchlaufen werden
    ELSE
      Teiler = i
    END IF
  NEXT i
  PRINT
  PRINT
  PRINT Teiler; "ist der größte gemeinsame Teiler von"; a; "und"; b
  LOCATE 20, 5
  INPUT "Weitere Aufgabe rechnen? j/n :", tastaturabrage$
  LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

2. Wertetabellen berechnen und ausgeben

2.1 Multiplikationstabelle

Aufgabe:

Erstelle ein Programm, daß die Multiplikationstabelle für alle Produkte der geraden und ungeraden Zahlen von 1-20 erstellt.

	2	4	6	8	10	12	14	16	18	20
1	2	4	6	8	10	12	14	16	18	20
3	6	12	18	24	30	36	42	48	54	60
5	10	20	30	40	50	60	70	80	90	101

_____Das Ergebnis soll in einer Tabelle ausgegeben werden._____

Programmlisting:

REM Wertetabellen Berechnung und Ausgabe: ➔ Programm 2.1

REM Die folgende Tabelle wird ohne Verwendung von PRINT USING als
REM sauber strukturierte Tabelle ausgegeben.
REM Man vergleiche den Aufwand mit Aufgabe 7 !

```
DO
REDIM zeile$(90)
CLS
PRINT "*****"
PRINT "*      Ausgabe der Multiplikationstabelle aller geraden      *"
PRINT "*                               mit ungeraden Zahlen                               *"
PRINT "*****"
PRINT
PRINT
PRINT " Die Zeilen können von 1 bis 30 beliebig vorgeben werden."
PRINT
INPUT "Geben Sie die Anzahl der Zeilen vor! :", zeilen
IF zeilen > 30 THEN
    COLOR 20 '➔ ändert Farbattribut auf rot/blinkend
    PRINT "Mehr als 30 Zeilen sind nicht möglich."
ELSE
CLS
PRINT "*****"
PRINT "*      Ausgabe der Multiplikationstabelle aller geraden      *"
PRINT "*                               mit ungeraden Zahlen von 1 bis"; zeilen; "                               *"
PRINT "*****"
PRINT "+-----+"
PRINT "| * | 2  4  6  8 10 12 14 16 18 20|"
PRINT "| | -+-----+ |"
```

```
_____
_____
_____
_____
_____
_____
_____
_____
_____
```

```
FOR i = 1 TO zeilen STEP 2
FOR j = 2 TO 20 STEP 2
IF j = 20 THEN
IF (j - 1) * i > 99 THEN
zeile$(i) = zeile$(i) + " " + STR$(i * j) + " | "
ELSE
IF i * j < 99 THEN
zeile$(i) = zeile$(i) + STR$(i * j) + " | "
ELSE
zeile$(i) = zeile$(i) + STR$(i * j) + " | "
END IF
END IF
ELSE
IF i * j < 10 THEN
zeile$(i) = zeile$(i) + STR$(i * j) + " | "
END IF
IF i * j < 99 AND i * j > 9 THEN
IF (j + 1) * i > 99 THEN
zeile$(i) = zeile$(i) + STR$(i * j) + " | "
ELSE
zeile$(i) = zeile$(i) + STR$(i * j) + " | "
END IF
END IF
IF i * j > 99 THEN
IF (j - 1) * i > 99 THEN
zeile$(i) = zeile$(i) + " " + STR$(i * j) + " | "
ELSE
zeile$(i) = zeile$(i) + STR$(i * j) + " | "
END IF
END IF
END IF
NEXT j
IF i < 10 THEN
PRINT "| " + STR$(i) + " | " + zeile$(i)
ELSE
PRINT "| " + STR$(i) + " | " + zeile$(i)
END IF
NEXT i
PRINT "-----1"
END IF
COLOR 7, 0 '➔ Setzt auf ursprüngliche Farben zurück
LOCATE 24, 5 '➔ Lokalisiert Zeile 24 und Spalte 5 für nächsten
Befehl
INPUT "Noch mal durchrechnen? j/n :", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

2.2 Sinustabelle

Aufgabe:

Gesucht ist ein Programm, welches Sinuswerte in einer übersichtlichen Form ausgibt.

Winkel		Sinus	Winkel		Sinus
0°	=	0,0000	1°	=	0,0174
2°	=	0,0349	3°	=	0,0523
4°	=	5°	=
.....				

Hinweis:

Verwende den Befehl - PRINT USING "#.####" - für eine einfache und übersichtliche Ausgabe.

Programmlisting:

REM Wertetabellen Berechnung und Ausgabe: → Programm 2.2

REM In diesem Programm wird gezeigt wie unter Verwendung von PRINT USING
REM spezielle Formatierungen von Zahlen ausgegeben werden können.

DO

Pi = 3.1415 '→ Pi als Konstant definieren

CLS

PRINT "*****"

PRINT "* Ausgabe der Winkel - Sinustabelle von 0° bis 90° *"

PRINT "* in Schritten von 1° *"

PRINT "*****"

PRINT

PRINT "Winkel=sin Winkel=sin Winkel=sin Winkel=sin Winkel=sin Winkel=sin"

PRINT "-----"

halt = 0

FOR i = 1 TO 15

mitzaehlen = 0

FOR winkel = halt TO 90

mitzaehlen = mitzaehlen + 1

sinus = SIN(2 * Pi * winkel / 360)

IF mitzaehlen < 7 THEN '→ sieben Werte pro Zeile ausgeben

halt = halt + 1 '→ Letzter Winkel wird zwischen gespeichert

PRINT USING "##°="; winkel; '→ Zahlenausgabe mit zwei Stellen und Grad

PRINT USING "#.#### | "; sinus; '→ Zahlenausgabe mit vier Nachkommastellen

END IF

NEXT winkel

PRINT

NEXT i

COLOR 20

PRINT "... und der sin von 90° ist: na, na , na richtig 1 :-)"

COLOR 7, 0 '→ Setzt auf ursprüngliche Farben zurück

LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl

INPUT "Noch mal durchrechnen? j/n :", tastaturabrage\$

LOOP UNTIL tastaturabrage\$ = "n" OR tastaturabrage\$ = "N"

END

3. Grenzwerte von Folgen, berechnen von Reihenwerten

3.1 Grenzwert einer gegebenen Folge

Aufgabe:

Gegeben sei die Folge :
$$\lim_{n \rightarrow \infty} \left(\frac{4 + 2n - 2n^2}{2n^2 - 2} \right)$$

Schreibe nun ein Programm zur Berechnung des Grenzwertes. Das Programm soll abgebrochen werden, wenn der Wert -1.499 erreicht ist. Mit dem Befehl - PRINT USING "##.#####" sind die erreichten Werte in einer Tabelle übersichtlich auszugeben.

Programmlisting:

REM Grenzwerte und Reihenwerte: --> Programm 3.1

REM In diesem Programm wird gezeigt wie unter Verwendung von PRINT USING
REM eine Grenzwerttabelle ausgegeben werden kann.

DO

CLS

PRINT "*****"

PRINT "* Ermittlung des Grenzwertes und Ausgabe in einer *"

PRINT "* Tabelle unter Verwendung des Befehls PRINT USING *"

PRINT "*****"

PRINT

FOR n = 2 TO 2000

f = (4 + 2 * n - 3 * n ^ 2) / (2 * n ^ 2 - 2)

IF f > -1.47938 THEN

PRINT USING "##.#####" f;

END IF

NEXT n

PRINT

PRINT

PRINT " . zwischen -1.47938 und -1.49894 ."

PRINT " . wurde die Bildschirmausgabe ."

PRINT " . unterbrochen ."

PRINT

FOR n = 2 TO 2000

f = (4 + 2 * n - 3 * n ^ 2) / (2 * n ^ 2 - 2)

IF f < -1.49895 AND f > -1.499 THEN

PRINT USING "##.#####" f;

END IF

NEXT n

PRINT

COLOR 20 '➔ Nächster Befehl wird rot/blinkend ausgegeben

PRINT "bei n=-1.499 wurde abgebrochen. ";

COLOR 7, 0 '➔ Setzt auf ursprüngliche Farben zurück

PRINT "Die Schleife wurde" + STR\$(n) + "-mal durchlaufen."

LOCATE 24, 5 '➔ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl

INPUT "Noch mal durchrechnen? j/n : ", tastaturabrage\$

LOOP UNTIL tastaturabrage\$ = "n" OR tastaturabrage\$ = "N"

END

3.2 Reihenentwicklung

Aufgabe:

Für **e** gilt die Reihenentwicklung:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!}$$

Andere Darstellungen für **e** sind:

$$e = \lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^{-n} \quad \text{und} \quad e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^{-n}$$

Mit diesen drei Möglichkeiten ist **e** zu berechnen. Die Berechnung ist abzubrechen, sobald der jeweilige Näherungswert gegenüber dem Taschenrechnerwert (2,71828182846) eine Abweichung $< 10^6$ aufweist. Um einen Konvergenzvergleich für die drei Verfahren durchführen zu können, ist die Anzahl der Iterationsschritte zu speichern und mit auszudrucken.

	Wert	Schrittzahl
Reihe
lim(-n)
lim(n)

Programmlisting:

REM Grenzwerte und Reihenwerte: → Programm 3.2

REM In diesem Programm wird gezeigt wie unter Verwendung von PRINT USING
REM spezielle Formatierungen von Zahlen ausgegeben werden können.

```
REM EXP(1) = 2.718281829 = e → Euler'sche Zahl
Z = 1
W = 1
DO
CLS
PRINT "*****"
PRINT "* Ermittlung des Wertes e über Reihenentwicklung *"
PRINT "* sowie durch Grenzwertberechnung von oben und unten *"
PRINT "*****"
PRINT
PRINT
COLOR 20 '→ Nächster Befehl wird rot/blinkend ausgegeben
PRINT "Die Berechnung kann je nach Rechner einige Zeit dauern";
COLOR 7, 0 '→ Setzt auf ursprüngliche Farben zurück
PRINT
PRINT
FOR i = 1 TO 15 '→ über Reihenentwicklung
IF EXP(1) - W > .000001 THEN
Z = Z * i
E = 1 / Z
W = W + E
N = N + 1
END IF
NEXT i
PRINT
PRINT "Wert Schrittzahl"
PRINT "Reihe e="; W; " "; N
PRINT
PRINT
F = (1 - 1 / 2) ^ (-2)
FOR i = 2 TO 100000 '→ über Grenzwert von unten
IF F - EXP(1) >= .0001 THEN
F = (1 - 1 / i) ^ (-i)
S = S + 1
END IF
NEXT i
```

```
PRINT "lim(-n) e="; F; " "; S
FOR i = 2 TO 100000 '→ über Grenzwert von unten
IF EXP(1) - C >= .0001 THEN
C = (1 + 1 / i) ^ (i)
K = K + 1
END IF
NEXT i
PRINT "lim(n) e="; F; " "; K
PRINT
CLS
PRINT "*****"
PRINT "* Ermittlung des Wertes e über Reihenentwicklung *"
PRINT "* sowie durch Grenzwertberechnung von oben und unten *"
PRINT "*****"
PRINT
PRINT
PRINT
PRINT "Wert Schrittzahl"
PRINT "Reihe e="; W; " "; N
PRINT
PRINT
PRINT "lim(-n) e="; F; " "; S
PRINT "lim(n) e="; F; " "; K
PRINT
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten
Befehl
INPUT "Noch mal durchrechnen? j/n :", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

3.3 Berechnung von Reihenwerten

Aufgabe:

Gegeben seien folgende Reihenwerte:

$$\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \frac{1}{4 \cdot 5} + \dots = 1$$

Setze nun die Reihenwerte in ein Programm um. Wenn die Differenz < 0.001 ist, soll das Programm abbrechen und den gerundeten Wert ausgeben.

Hinweis:

Eine Zahl wird mit dem Befehl - CINT(X) - aufgerundet.

Programmlisting:

REM Grenzwerte und Reihenwerte: → Programm 3.3

```
Y = 1
DO
  CLS
  PRINT "*****"
  PRINT "*" Ermittlung des Wertes e über Reihenentwicklung *
  PRINT "*" Der Grenzwert wird dann aufgerundet durch CINT ausgegeben *"
  PRINT "*****"
  PRINT
  PRINT " 1      1      1      1      1      "
  PRINT " ---- + ---- + ---- + ---- + ... + ----- = e"
  PRINT "1 * 2  2 * 3  3 * 4  4 * 5      (n - 1) * n  "
  PRINT
  COLOR 20 '→ Nächster Befehl wird rot/blinkend ausgegeben
  PRINT "Die Berechnung kann je nach Rechner einige Zeit dauern";
  COLOR 7, 0 '→ Setzt auf ursprüngliche Farben zurück
  PRINT
  PRINT
  FOR i = 2 TO 100000
    IF 1 - W >= .00001 THEN
      F = 1 / (Y * i)
      W = F + W
      Y = Y + 1
    END IF
  NEXT i
  CLS
  PRINT "*****"
  PRINT "*" Ermittlung des Wertes e über Reihenentwicklung *
  PRINT "*****"
  PRINT
  PRINT
  PRINT
  PRINT " 1      1      1      1      1      "
  PRINT " ---- + ---- + ---- + ---- + ... + ----- = e"
  PRINT "1 * 2  2 * 3  3 * 4  4 * 5      (n - 1) * n  "
  PRINT
  PRINT
  PRINT
  PRINT "Die Reihe wurde"; Y; "- mal entwickelt."
  PRINT
  PRINT USING "Der letzte Wert war dabei X = #.#####"; W
  PRINT
  PRINT "Die Reihe konvergiert also gegen e ="; CINT(W)
  LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
  NPUT "Noch mal durchrechnen? j/n :", tastaturabrage$
  LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

3.4 Rechnen mit einer Zahlenreihe

Aufgabe:

Gegeben ist die Zahlenreihe: $1^3+3^3+5^3+....$

Es soll nun ein Programm geschrieben werden, das die Zahlenwerte in einer Tabelle darstellt. Ist eine Zahl größer als 200000, so soll das Programm gestoppt werden. Gesucht ist die Summe der Zahlenfolge, die Anzahl der Zahlen und der arithmetische Mittelwert. Stelle das Ergebnis in einer übersichtlichen Form dar.

Programmlisting:

REM Grenzwerte und Reihenwerte: --> Programm 3.4

DO

CLS

PRINT "*****"

PRINT "* Ermittlung der Summe sobald k^3 größer 200000 und anschließende *"

PRINT "* Ausgabe des Mittelwertes und der Anzahl der berechneten Zahlen *"

PRINT "*****"

PRINT

PRINT

PRINT " 3 3 3 3 n 3"

PRINT "1 + 3 + 5 + + (2n-1) = S (2k - 1) "

PRINT " k = 1"

PRINT

PRINT

FOR i = 1 TO 100 STEP 2

IF F <= 200000 THEN

F = i ^ 3

W = W + F

Y = Y + 1

PRINT USING "##### | "; F;

END IF

NEXT i

PRINT

PRINT

PRINT "Die Anzahl der berechneten Zahlen beträgt: "; Y

PRINT

PRINT "Die Summe der Zahlenfolge beträgt : "; W

PRINT

PRINT "Das arithmetische Mittel beträgt : "; W / Y

LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl

INPUT "Noch mal durchrechnen? j/n :", tastaturabrage\$

LOOP UNTIL tastaturabrage\$ = "n" OR tastaturabrage\$ = "N"

END

4. Iterationsverfahren zur Nullstellenbestimmung von Funktionen

4.1 Intervallhalbierungsmethode

Aufgabe:

Es ist eine Funktion $f(x)$ gegeben, die im Intervall $[a, b]$ genau einmal die x -Achse schneidet. Dieser Punkt (Nullstelle) soll durch die Intervallhalbierungsmethode iterativ so lange angenähert werden, bis die vorgegebene Genauigkeit auf d Stellen erreicht wird.

Verfahren:

Das Intervall wird in 2 gleich große Teilintervalle geteilt. Es werden die Funktionswerte von beiden Teilintervallgrenzen berechnet. Das Teilintervall, bei dem die Funktionswerte der Teilintervallgrenzen ihr Vorzeichen wechseln, beinhaltet die Nullstelle und wird als neues Ausgangsintervall verwendet.

Die Anzahl n der Iterationen berechnet sich aus:

$$n > 1 + \frac{d + \log(b - a)}{\log(2)}$$

Hinweis:

Das Vorzeichen einer Zahl x wird durch den Befehl $\text{SGN}(X)$ erfragt.

Programmlisting:

REM Iterationsverfahren zur Nullstellenbestimmung: → Programm 4.1

```
DEF FNF (x) = (x + 2) * EXP(x - 1) '→ Definition der Funktion
DO
CLS
PRINT "*****"
PRINT "* Durch Intervallhalbierung wird eine Nullstelle einer Funktion *"
PRINT "* mit der vorgegebenen Genauigkeit d ermittelt *"
PRINT "*****"
PRINT
PRINT
PRINT
PRINT " (x-1)"
PRINT "Die Funktion lautet: F(x) = x * e^"
PRINT
INPUT " Intervallgrenze: a=", a
INPUT " Intervallgrenze: b=", b
INPUT "Genauigkeit der Nullstelle auf d Stellen: d=", d
IF a > b THEN
c = a '→ Vertauschung der Intervallgrenzen:
a = b '→ Aufgrund der Abarbeitung des Programms
b = c '→ muß a immer kleiner als b sein
END IF
IF a = b THEN
COLOR 20 '→ Nächster Bildschirmbefehl wird rot/blinkend ausgegeben
PRINT
PRINT "Falsche Intervallgrenzen!! → Intervallgrenzen dürfen nicht gleich sein! "
COLOR 7, 0 '→ Setzt auf ursprüngliche Farben zurück
ELSE
N = 2 + (d + .434 * LOG(b - a)) / LOG(2) '→ Iterationstiefe, LOG=natürlicher Logarithmus
FOR I = 1 TO N
mitte = (a + b) / 2
Flinks = FNF(a)
Fmitte = FNF(mitte)
IF Fmitte = 0 THEN
EXIT FOR '→ For-Schleife wird abgebrochen
ELSE
a = mitte
END IF
Frechts = FNF(b)
IF SGN(Flinks) <> SGN(Fmitte) THEN '→ SGN=Vorzeichen, <> = ungleich
b = mitte
END IF
NEXT I
PRINT
PRINT
PRINT
PRINT "Die Nullstelle liegt bei x = "; mitte
PRINT
END IF
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten
Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabfrage$
LOOP UNTIL tastaturabfrage$ = "n" OR tastaturabfrage$ = "N"
END
```

4.2 Sekantenmethode

Aufgabe:

Die Aufgabe aus 4.1 soll mit Hilfe der Sekantenmethode gelöst werden. Die Iteration soll so lange wiederholt werden, bis die Differenz der beiden letzten Näherungen kleiner als ϵ ist.

Verfahren:

Das Intervall (L,R) wird wieder in 2 Teilintervalle zerlegt. Der gemeinsame Grenzpunkt m der beiden Teilintervalle liegt jedoch an der Stelle, an der die durch $f(L)$ und $f(R)$ gehende Gerade die x-Achse schneidet.

$$m = \frac{R - f(R) \cdot (R - L)}{f(R) - f(L)}$$

Programmlisting:

REM Iterationsverfahren zur Nullstellenbestimmung: → Programm 4.2

```
DEF FNF (x) = (x + 2) * EXP(x - 1) '→ Definition der Funktion
DO
CLS
PRINT "*****"
PRINT "* Mitteltes Sekantenmethode wird eine Nullstelle einer Funktion *"
PRINT "* mit der vorgegebenen Genauigkeit d ermittelt *"
PRINT "*****"
PRINT
PRINT
PRINT
PRINT " (x-1)"
PRINT "Die Funktion lautet: F(x) = x * e^"
PRINT
INPUT " Intervallgrenze: a=", a
INPUT " Intervallgrenze: b=", b
INPUT "Genauigkeit der Nullstelle auf d Stellen: d=", d
IF a > b THEN
c = a '→ Vertauschung der Intervallgrenzen:
a = b '→ Aufgrund der Abarbeitung des Programms
b = c '→ muß a immer kleiner als b sein
END IF
IF a = b THEN
COLOR 20 '→ Nächster Bildschirmbefehl wird rot/blinkend ausgegeben
PRINT
PRINT "Falsche Intervallgrenzen!! → Intervallgrenzen dürfen nicht gleich sein! "
COLOR 7, 0 '→ Setzt auf ursprüngliche Farben zurück
ELSE
Wiederh = -1
M = d + 33
Anz = 0
DO
Anz = Anz + 1 '→ Anzahl der Iterationen
Flinks = FNF(a)
Frechts = FNF(b)
Malt = M
M = b - Frechts * (a - b) / (Frechts - Flinks) '→ Grenzpunkt
Fmitte = FNF(M)
Frechts = FNF(b)
IF SGN(Flinks) <> SGN(Fmitte) THEN '→ wenn Vorzeichenwechsel
b = M
Wiederh = (ABS(M - Malt) > d) '→ Prüfung ob Genauigkeit reicht
END IF
IF Fmitte = 0 THEN
EXIT DO '→ DO-Schleife wird abgebrochen
ELSE
a = M
END IF
Wiederh = (ABS(M - Malt) > d) '→ Prüfung ob Genauigkeit reicht
LOOP UNTIL Wiederh = 0 '→ Wiederhole solange bis Wiederh = 0
```

```
PRINT
PRINT
PRINT
PRINT "Die Nullstelle liegt bei x = "; M
PRINT
END IF
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten
Befehl
INPUT "Noch mal durchrechnen? j/n :", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```


4.3 Newtonsches Iterationsverfahren

Aufgabe:

Eine Nullstelle einer Funktion soll mit Hilfe des Newtonschen Iterationsverfahrens so lange angenähert werden, bis die Differenz der letzten beiden Näherungen kleiner als ϵ ist.

Hinweis:

Die Näherungsformel lautet:

$$X(k+1) = X(k) - \frac{f(x(k))}{f'(x(k))}$$

Programmlisting:

REM Iterationsverfahren zur Nullstellenbestimmung: → Programm 4.3

DEF FNF1 (x) = (x) * LOG(x) - 1 '→ Definition der Funktion f(x)

DEF FNF2 (x) = LOG(x) + 1 '→ Ableitung der Funktion

DO

CLS

PRINT "*****"

PRINT "* Mittels Newtonschen Iterationsverfahren wird eine Nullstelle *

PRINT "* einer Funktion mit der vorgegebenen Genauigkeit ermittelt *"

PRINT "*****"

PRINT

PRINT

PRINT

PRINT

PRINT "Die Funktion lautet: f(x) = x * log(x) - 1"

PRINT

INPUT "Abweichung der letzten Näherung auf wieviel Stellen genau: d=", d

INPUT "Eingabe der Startwertes : xn=", xn

Wiederh = -1

Anz = 0

DO

Anz = Anz + 1 '→ Anzahl der Iterationen

xa = xn

xn = xa - FNF1(xa) / FNF2(xa) '→ Neue Näherung

Wiederh = (ABS(xn - xa) > d) '→ Prüfung ob Genauigkeit reicht

LOOP UNTIL Wiederh = 0 '→ Wiederhole solange bis Wiederh = 0

PRINT

PRINT

PRINT

PRINT

PRINT "Die Nullstelle liegt bei x = "; xn

PRINT

LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl

INPUT "Noch mal durchrechnen? j/n :", tastaturabrage\$

LOOP UNTIL tastaturabrage\$ = "n" OR tastaturabrage\$ = "N"

END

5. Eindimensionale Felder

5.1 Primzahlen bestimmen (Methode nach Erathostenes)

Aufgabe:

Wir schreiben zuerst alle Zahlen von 1 bis zu einer selbst festgelegten Grenze nacheinander auf, nehmen die Zahl 2 und dividieren eine Zahl nach der anderen durch 2. Ist eine Zahl ein Vielfaches von 2 wird sie gestrichen. Haben wir alle Zahlen dividiert, nehmen wir die Zahl 3 und wiederholen den obigen Vorgang. Als nächste Zahl nehmen wir die Zahl, die bis jetzt noch nicht gestrichen wurde, d.h. die Zahl 5 usw. Zum Schluß bleiben die Primzahlen als unteilbare Zahl übrig.

Hinweis:

Eine Zahl wird gestrichen, indem wir sie zu Null setzen.

Programmlisting:

REM Eindimensionale Felder: → Programm 5.1

DECLARE SUB seitenweiseBildschirmausgabe () '→ Unterprogramm deklarieren

DIM SHARED aufpassen, substart '→ Diese Variablen sind global verfügbar

```
DO
aufpassen = 0
substart = 0
CLS
PRINT "*****"
PRINT "* Bestimmung von Primzahlen nach der Methode von Erathostenes *"
PRINT "*****"
PRINT
PRINT
INPUT "Bis zu welcher Zahl sollen die Primzahlen bestimmt werden: n=", n
REDIM SHARED z(n)
FOR i = 1 TO n
    z(i) = i
NEXT i
z(1) = 0
FOR j = 2 TO SQR(n)
    g = n / j
    FOR k = 2 TO g
        z(j * k) = 0
    NEXT k
NEXT j
FOR i = 1 TO n
    IF z(i) <> 0 THEN
        IF i = 1 THEN
            'lasse ausgeben aus, da erste Zahl immer obere Grenze + 1 ist
        ELSE
            IF n > 1500 THEN
                'seitenweiseBildschirmausgabe → man probiere ohne Unterprogramm
                seitenweiseBildschirmausgabe 'Unterprogramm zur besseren Ausgabe
            END IF
            IF substart = 0 THEN '→ wird nur ausgeführt wenn Unterprogramm
                PRINT USING "#### "; z(i); '→ nicht aufgerufen wurde
            END IF
        END IF
    END IF
NEXT i
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n "; tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

```
SUB seitenweiseBildschirmausgabe
SHARED z(), n
IF substart = 0 THEN
    aufpassen = 0
    FOR i = 2 TO n
        IF z(i) <> 0 THEN
            IF mitzaehlen = 0 THEN
                CLS
                PRINT
                "*****"
                PRINT "* Bestimmung von Primzahlen nach der Methode von Erathostenes *"
                "*****"
                PRINT
                "*****"
                PRINT
                END IF
            mitzaehlen = mitzaehlen + 1
            PRINT USING "#### "; z(i);
            IF mitzaehlen = 288 THEN
                mitzaehlen = 0
                LOCATE 23, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
                COLOR 20
                PRINT "Leertaste für nächste Seite!"
                COLOR 7, 0
                DO
                    LOOP UNTIL INKEY$ = CHR$(32)
                END IF
            END IF
        END IF
    NEXT i
    substart = 1
END IF
END SUB
```

5.2 Sortieren durch direkten Vergleich (Bubblesort)

Aufgabe:

Die Routine sucht sich zuerst die kleinste Zahl aus der Menge, als nächstes die zweitkleinste usw. bis allen Positionen ihre Zahl zugeordnet wurde.

unsortiert

9 6	1 3	3 9	9 3	5 1	5 5	6 8	5	7 7	4 0
5	9 6	3 9	9 3	5 1	5 5	6 8	1 3	7 7	4 0
5	1 3	9 6	9 3	5 1	5 5	6 8	3 9	7 7	4 0
5	1 3	3 9	4 0	9 6	9 3	6 8	5 5	7 7	5 1
5	1 3	3 9	4 0	5 1	9 6	9 3	6 8	7 7	5 5
5	1 3	3 9	4 0	5 1	5 5	9 6	9 3	7 7	6 8
5	1 3	3 9	4 0	5 1	5 5	6 8	9 6	9 3	7 7
5	1 3	3 9	4 0	5 1	5 5	6 8	7 7	9 6	9 3
5	1 3	3 9	4 0	5 1	5 5	6 8	7 7	9 3	9 6

sortiert

Hinweis:

Tauschen von zwei Variablen.

H = Z(i) : Z(i) → H

Z(i) = Z(j) : Z(j) → Z(i)

Z(j) = H : H → Z(j)

Programmlisting:

REM Eindimensionale Felder: → Programm 5.2

DECLARE SUB seitenweiseBildschirmAusgabe () '→ Unterprogramm deklarieren

DIM SHARED aufpassen, substart '→ Diese Variablen sind global verfügbar

DO

aufpassen = 0

substart = 0

CLS

PRINT "*****"

PRINT "* Sortieren durch direkten Vergleich (Bubblesort) *"

PRINT "*****"

PRINT

PRINT

INPUT "Wieviel Zahlen sollen sortiert werden: Anzahl = ", n

REDIM SHARED z(n)

FOR i = 1 TO n

PRINT i; " ";

INPUT "Zahl = ", z(i)

NEXT i

FOR i = 1 TO n - 1 '→ sortieren

FOR j = i + 1 TO n

IF z(i) > z(j) THEN

h = z(i)

z(i) = z(j)

z(j) = h

END IF

NEXT j

NEXT i

CLS

PRINT "*****"

PRINT "* Sortieren durch direkten Vergleich (Bubblesort) *"

PRINT "*****"

PRINT

PRINT

FOR i = 1 TO n

IF n > 280 THEN

'seitenweiseBildschirmAusgabe → man probiere ohne Unterprogramm aus

seitenweiseBildschirmAusgabe 'Unterprogramm zur besseren Ausgabe

END IF

IF substart = 0 THEN '→ wird nur ausgeführt wenn Unterprogramm

PRINT USING "#### "; z(i); '→ nicht aufgerufen wurde

END IF

NEXT i

LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl

INPUT "Nochmal durchrechnen? j/n : ", tastaturabfrage\$

LOOP UNTIL tastaturabfrage\$ = "n" OR tastaturabfrage\$ = "N"

END

SUB seitenweiseBildschirmAusgabe

SHARED z(), n

IF substart = 0 THEN

aufpassen = 0

FOR i = 1 TO n

IF mitzaehlen = 0 THEN

CLS

PRINT

"*****"

PRINT "* Sortieren durch direkten Vergleich (Bubblesort)

*****"

PRINT

"*****"

PRINT

END IF

mitzaehlen = mitzaehlen + 1

PRINT USING "#### "; z(i);

IF mitzaehlen = 288 THEN

mitzaehlen = 0

LOCATE 23, 5 '→ Lokalisiert Zeile 24 und Spalte 5

COLOR 20

PRINT "Leertaste für nächste Seite!"

COLOR 7, 0

DO

LOOP UNTIL INKEY\$ = CHR\$(32)

END IF

NEXT i

substart = 1

END IF

END SUB

5.3 Sortieren durch benachbarten Vergleich

Aufgabe:

Dieses Verfahren ist dem in 5.3 Verfahren durch direkten Vergleich ähnlich. Hier werden die Elemente von hinten nach sortiert und der Größe nach ihren Positionen zugeordnet.

unsortiert:

20	6	98	87	10	8	70	15	95	98
6	20	87	10	8	70	15	95	98	98
6	20	10	8	70	15	87	95	98	98
6	10	8	20	15	70	87	95	98	98
6	8	10	15	20	70	87	95	98	98
6	8	10	15	20	70	87	95	98	98

sortiert:

Hinweis:

Die Variable IC ist eine boolsche Variable, die lediglich die Werte 0 und 1 annimmt. Ist ihr Wert 1, kennzeichnet dies ein nochmaliges Durchlaufen der Schleife, weil noch nicht alle Elemente sortiert sind.

Programmlisting:

```
REM Eindimensionale Felder: → Programm 5.3
DECLARE SUB seitenweiseBildschirmAusgabe () '→ Unterprogramm deklarieren
DIM SHARED aufpassen, substart '→ Diese Variablen sind global verfügbar

DO
aufpassen = 0
substart = 0
CLS
PRINT "*****"
PRINT "*          Sortieren durch benachbarten Vergleich          *"
PRINT "*****"
PRINT
PRINT
INPUT "Wieviel Zahlen sollen sortiert werden: Anzahl = ", n
REDIM SHARED z(n)
FOR i = 1 TO n
    PRINT i; ". ";
    INPUT "Zahl = ", z(i)
NEXT i
DO
ic = 0
FOR i = 1 TO n - 1 '→ sortieren
    IF z(i) > z(i + 1) THEN
        h = z(i)
        z(i) = z(i + 1)
        z(i + 1) = h
        ic = 1
    END IF
NEXT i
LOOP UNTIL ic = 0
CLS
PRINT "*****"
PRINT "*          Sortieren durch benachbarten Vergleich          *"
PRINT "*****"
PRINT
PRINT
FOR i = 1 TO n
    IF n > 280 THEN
        'seitenweiseBildschirmAusgabe → man probiere ohne Unterprogramm aus
        seitenweiseBildschirmAusgabe 'Unterprogramm zur besseren Ausgabe
    END IF
    IF substart = 0 THEN '→ wird nur ausgeführt wenn Unterprogramm
        PRINT USING "#### "; z(i); '→ nicht aufgerufen wurde
    END IF
NEXT i
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabfrage$
LOOP UNTIL tastaturabfrage$ = "n" OR tastaturabfrage$ = "N"
END

SUB seitenweiseBildschirmAusgabe
SHARED z(), n
IF substart = 0 THEN
    aufpassen = 0
    FOR i = 1 TO n
        IF mitzaehlen = 0 THEN
            CLS
            PRINT "*****"
            PRINT "*          Sortieren durch benachbarten Vergleich          *"
            PRINT "*****"
            PRINT
            END IF
            mitzaehlen = mitzaehlen + 1
            PRINT USING "#### "; z(i);
            IF mitzaehlen = 288 THEN
                mitzaehlen = 0
                LOCATE 23, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten
                Befehl
                COLOR 20
                PRINT "Leertaste für nächste Seite!"
                COLOR 7, 0
                DO
                    LOOP UNTIL INKEY$ = CHR$(32)
                END IF
            END IF
        NEXT i
        substart = 1
    END IF
END SUB
```

5.4 Sortieren durch Einfügen

Aufgabe:

Die Zahlen werden nacheinander aus der unsortierten Menge genommen und in die sortierte eingefügt.
Das gesamte Zahlenfeld

68	5	78	35	22	85	49	8	7
5	68	78	35	22	85	49	8	7
5	68	78	35	22	85	49	8	7
5	35	68	78	22	85	49	8	7
5	22	35	68	78	85	49	8	7
5	22	35	68	78	85	49	8	7
5	22	35	49	68	78	85	8	7
5	8	22	35	49	68	78	85	7
5	7	8	22	35	49	68	78	85

Das Anwachsende der sortierten Liste

68								
5	68							
5	68	78						
5	35	68	78					
5	22	35	68	78				
5	22	35	68	78	85			
5	22	35	49	68	78	85		
5	8	22	35	49	68	78	85	
5	7	8	22	35	49	68	78	85

Hinweis:

Die Variable J kennzeichnet die Position der einzusortierenden Zahl in der sortierten Liste. Die Variable H übernimmt das einzusortierende Element, damit es nicht beim Einfügen überschrieben wird. Die FOR - NEXT-Schleife der Variablen K schiebt die Elemente der sortierten Liste um eine Position nach hinten, um das neue Element einzusortieren.

Programmlisting:

REM Eindimensionale Felder: → Programm 5.4

DECLARE SUB seitenweiseBildschirmAusgabe () '→ Unterprogramm deklarieren

DIM SHARED aufpassen, substart '→ Diese Variablen sind global verfügbar

```
DO
aufpassen = 0
substart = 0
CLS
PRINT "*****"
PRINT "*"          Sortieren durch Einfügen          "*"
PRINT "*****"
PRINT
PRINT
INPUT "Wieviel Zahlen sollen sortiert werden: Anzahl = ", n
REDIM SHARED z(n)
FOR i = 1 TO n
    PRINT i; ". ";
    INPUT "Zahl = ", z(i)
NEXT i
FOR i = 2 TO n          '→ sortieren
    j = 1
    DO
        IF z(i) > z(j) THEN
            j = j + 1
        END IF
    LOOP UNTIL z(i) <= z(j)
    h = z(i)
    FOR k = i - 1 TO j STEP -1
        z(k + 1) = z(k)
    NEXT k
    z(j) = h
NEXT i
CLS
PRINT "*****"
PRINT "*"          Sortieren durch Einfügen          "*"
PRINT "*****"
PRINT
PRINT
FOR i = 1 TO n
    IF n > 280 THEN
        'seitenweiseBildschirmAusgabe → man probiere ohne Unterprogramm aus
        seitenweiseBildschirmAusgabe 'Unterprogramm zur besseren Ausgabe
    END IF
    IF substart = 0 THEN          '→ wird nur ausgeführt wenn Unterprogramm
        PRINT USING "#### "; z(i); '→ nicht aufgerufen wurde
    END IF
NEXT i
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabfrage$
```

```
SUB seitenweiseBildschirmAusgabe
SHARED z(), n
IF substart = 0 THEN
    aufpassen = 0
    FOR i = 1 TO n
        IF mitzaehlen = 0 THEN
            CLS
            PRINT "*****"
            PRINT "*"          Sortieren durch Einfügen          "*"
            PRINT "*****"
            PRINT
            END IF
            mitzaehlen = mitzaehlen + 1
            PRINT USING "#### "; z(i);
            IF mitzaehlen = 288 THEN
                mitzaehlen = 0
                LOCATE 23, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten
                Befehl
                COLOR 20
                PRINT "Leertaste für nächste Seite!"
                COLOR 7, 0
                DO
                    LOOP UNTIL INKEY$ = CHR$(32)
                END IF
            NEXT i
            substart = 1
        END IF
    END SUB
```

```
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"  
END
```

5.5 Sortieren von Strings (am Beispiel 5.4)

Aufgabe:

Alle vorherigen Verfahren zum Sortieren von Zahlen funktionieren auch mit Stringvariablen. Hier ist jedoch darauf zu achten, daß der Computer die Umlaute ä, ö und ü nicht als ae, oe und ue einsortiert, sondern sie hinten an das Alphabet setzt (dasselbe bei ß). Weiterhin kommen alle Großbuchstaben vor die Kleinbuchstaben (siehe ASCII - Code).

Hinweis:

die Zahlenvariablen sind nun durch Stringvariablen ersetzt. Stringvariablen sind durch ein \$ gekennzeichnet.

Programmlisting:

REM Eindimensionale Felder: → Programm 5.5

DECLARE SUB seitenweiseBildschirmausgabe () '→ Unterprogramm deklarieren

DIM SHARED aufpassen, substart '→ Diese Variablen sind global verfügbar

```
DO
aufpassen = 0
substart = 0
CLS
PRINT "*****"
PRINT "*          Sortieren von Strings durch einfügen          *"
PRINT "*****"
PRINT
PRINT
INPUT "Wieviel Strings sollen sortiert werden: Anzahl = ", n
REDIM SHARED z$(n)
FOR i = 1 TO n
    PRINT i; ". ";
    INPUT "Zahl = ", z$(i)
NEXT i
FOR i = 2 TO n          '→ sortieren
    j = 1
    DO
        IF z$(i) > z$(j) THEN
            j = j + 1
        END IF
    LOOP UNTIL z$(i) <= z$(j)
    h$ = z$(i)
    FOR k = i - 1 TO j STEP -1
        z$(k + 1) = z$(k)
    NEXT k
    z$(j) = h$
NEXT i
CLS
PRINT "*****"
PRINT "*          Sortieren von Strings durch einfügen          *"
PRINT "*****"
PRINT
PRINT
FOR i = 1 TO n
    IF n > 280 THEN
        'seitenweiseBildschirmausgabe → man probiere ohne Unterprogramm aus
        seitenweiseBildschirmausgabe 'Unterprogramm zur besseren Ausgabe
    END IF
    IF substart = 0 THEN          '→ wird nur ausgeführt wenn Unterprogramm
        PRINT z$(i) + " "; '→ nicht aufgerufen wurde
    END IF
NEXT i
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

```
SUB seitenweiseBildschirmausgabe
SHARED z$( ), n
IF substart = 0 THEN
    aufpassen = 0
    FOR i = 1 TO n
        IF mitzaehlen = 0 THEN
            CLS
            PRINT "*****"
            PRINT "*          Sortieren von Strings durch einfügen          *"
            PRINT "*****"
            PRINT
        END IF
        mitzaehlen = mitzaehlen + 1
        PRINT z$(i) + " ";
        IF mitzaehlen = 288 THEN
            mitzaehlen = 0
            LOCATE 23, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten
            Befehl
            COLOR 20
            PRINT "Leertaste für nächste Seite!"
            COLOR 7, 0
            DO
                LOOP UNTIL INKEY$ = CHR$(32)
            END IF
        END IF
        substart = 1
    END IF
END SUB
```

6. Rechnen mit Vektoren

6.1 Addition von Vektoren - Winkel zwischen Vektoren

Aufgabe:

Berechne die Summe zweier Vektoren (im Raum), deren Skalarprodukt und der von ihnen eingeschlossene Winkel. Dazu werden die beiden Vektoren komponentenweise eingegeben, die Ergebnisse berechnet und dann komponentenweise wieder ausgegeben.

Hinweis:

Der Arcuscosinus von x lässt sich errechnen durch:

$$\arccos(x) = \frac{P}{2} - \arctan\left(\frac{X}{(1-x^2)^2}\right)$$

Programmlisting:

REM Rechnen mit Vektoren: ➔ Programm 6.1

DO

```
PI = 3.1415926#
CLS
PRINT "*****"
PRINT "* Berechnung des Skalarproduktes und des eingeschlossenen  *"
PRINT "* Winkels zweier Vektoren                                     *"
PRINT "*****"
PRINT
PRINT
INPUT "Eingabe Vektor 1: X: ", Ax
INPUT "      Y: ", Ay
INPUT "      Z: ", Az
PRINT
INPUT "Eingabe Vektor 2: X: ", Bx
INPUT "      Y: ", By
INPUT "      Z: ", Bz
SummeX = Ax + Bx
SummeY = Ay + By
SummeZ = Az + Bz
Skalarprodukt = Ax * Bx + Ay * By + Az * Bz
BetragA = SQR(Ax ^ 2 + Ay ^ 2 + Az ^ 2)
BetragB = SQR(Bx ^ 2 + By ^ 2 + Bz ^ 2)
Kos = Skalarprodukt / (BetragA * BetragB)
IF Kos = 1 THEN
    Winkel = 0
ELSE
    Winkel = 180 / PI * (PI / 2 - ATN(Kos / SQR(1 - Kos * Kos)))
END IF
PRINT
PRINT "-----"
PRINT
PRINT "      "; SummeX
PRINT "      Summe      "; SummeY
PRINT "      "; SummeZ
PRINT
PRINT "Skalarprodukt: "; Skalarprodukt
PRINT
PRINT "Winkel      "; Winkel; "°"
LOCATE 24, 5 '➔ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```


6.2 Vektorprodukt

Aufgabe:

Die Fläche des von zwei Vektoren im Raum aufgespannten Parallelogramms soll berechnet werden. Ebenfalls ist ein auf beiden Vektoren senkrecht stehender Normalenvektor zu bestimmen. Mit dem Vektorprodukt lassen sich beide Teilaufgaben lösen.

Programmlisting:

REM Rechnen mit Vektoren: ➔ Programm 6.2

DO

```
CLS
PRINT "*****"
PRINT "* Berechnung des Vektorproduktes und eines Normalenvektors *"
PRINT "* zweier Vektoren *"
PRINT "*****"
PRINT
PRINT
INPUT "Eingabe Vektor 1: X: ", Ax
INPUT " Y: ", Ay
INPUT " Z: ", Az
PRINT
INPUT "Eingabe Vektor 2: X: ", Bx
INPUT " Y: ", By
INPUT " Z: ", Bz
ProduktX = Ay - Bz
ProduktY = Az + Bx
ProduktZ = Ax + By
Betrag = SQR(ProduktX * ProduktX + ProduktY * ProduktY + ProduktZ * ProduktZ)
PRINT
PRINT "-----"
PRINT
PRINT " "; ProduktX
PRINT " Normalenvektor "; ProduktY
PRINT " "; ProduktZ
PRINT
PRINT
PRINT "Die Fläche des Parallelogramms beträgt: ", Betrag
LOCATE 24, 5 ➔ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabfrage$
LOOP UNTIL tastaturabfrage$ = "n" OR tastaturabfrage$ = "N"
END
```

6.3 Skalarprodukt

Aufgabe:

Das Volumen des von den Vektoren a, b, c aufgespannten Spats ist zu bestimmen.

Hinweis:

Das Volumen des Spats ist gleich dem Betrag des Spatprodukts. Unter dem Spatprodukt $(a \cdot b \cdot c)$ versteht man das Skalarprodukt aus dem Vektor a und dem aus den Vektoren b und c gebildeten Vektorprodukt $b \times c$:

$$(a \cdot b \cdot c) = a \cdot (b \times c)$$

Programmlisting:

REM Rechnen mit Vektoren: ➔ Programm 6.3

DO

```
CLS
PRINT "*****"
PRINT "*      Berechnung des Spatprodukts und des Volumens      *"
PRINT "*                        dreier Vektoren                        *"
PRINT "*****"
PRINT
PRINT
INPUT "Eingabe Vektor 1: X: ", Ax
INPUT "      Y: ", Ay
INPUT "      Z: ", Az
PRINT
INPUT "Eingabe Vektor 2: X: ", Bx
INPUT "      Y: ", By
INPUT "      Z: ", Bz
PRINT
INPUT "Eingabe Vektor 3: X: ", Cx
INPUT "      Y: ", Cy
INPUT "      Z: ", Cz
ProduktX = By - Cz '➔ Vektorprodukt B x C
ProduktY = Bz - Cx
ProduktZ = Bx - Cy
Skalarprodukt = Ax * ProduktX + Ay * ProduktY + Az * ProduktZ '➔ a * v
PRINT
PRINT "-----"
PRINT
PRINT
PRINT "Das Volumen des Spats beträgt: ", ABS(Skalarprodukt)
LOCATE 24, 5 '➔ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

7. Mehrdimensionale Felder

7.1 Addition zweier Matrizen

Aufgabe:

Zuerst werden die Beiden Matrizen **|A|** und **|B|** eingelesen und anschließend berechnet, indem die Matrizenelemente mit gleichen Indizes addiert und der neuen Matriz **|C|** zugeordnet werden. Das Ergebnis erscheint auf dem Bildschirm.

Hinweis:

Bei den folgenden Aufgaben ändern sich die Routinen “Matriz eingeben” und “Lösungsmatriz ausgeben” nicht. Sie bilden den Rahmen der Programme, bei denen lediglich die “Berechnung” wechselt.

Die Matriz wird zeilenweise eingegeben, so steht die Variable x für die Zeile und y für die Spalte. Durch DIM V(n) reserviert der Rechner für V(1), V(2), ..., V(n) Variablenplatz.

DIM V(n,m) ist eine zweidimensionale Matriz der Form:

$$\begin{array}{cccc} V(1,1) & V(1,2) & \cdots & V(1,m) \\ V(2,1) & V(2,2) & \cdots & V(2,m) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ V(n,1) & V(n,2) & \cdots & V(n,m) \end{array}$$

Programmlisting:

REM Mehrdimensionale Felder: → Programm 7.1

```
DO
CLS
PRINT "*****"
PRINT "*          Addition zweier Matrizens          *"
PRINT "*****"
PRINT
PRINT
PRINT "Matrizendimensionierung (Zeilen x Spalten)"
INPUT "Zeilenzahl : ", i
INPUT "Spaltenzahl : ", j
REDIM A(i, j), B(i, j)
PRINT
PRINT "Matriz 1"
PRINT
FOR x = 1 TO i
  FOR y = 1 TO j
    PRINT "("; x; ", "; y; ")";
    INPUT " = ", A(x, y)
  NEXT y
NEXT x
PRINT
PRINT "Matriz 2"
PRINT
FOR x = 1 TO i
  FOR y = 1 TO j
    PRINT "("; x; ", "; y; ")";
    INPUT " = ", B(x, y)
  NEXT y
NEXT x
REDIM C(i, j)
FOR x = 1 TO i
  FOR y = 1 TO j
    C(x, y) = A(x, y) + B(x, y)
  NEXT y
NEXT x
```

```
CLS
PRINT "*****"
PRINT "*          Addition zweier Matrizens          *"
PRINT "*****"
PRINT
PRINT
PRINT "Lösungsmatriz"
PRINT
PRINT
FOR x = 1 TO i
  FOR y = 1 TO j
    PRINT C(x, y);
  NEXT y
NEXT x
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten
Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

7.2 Matrizenaddition mit beliebig vielen Matrizen

Aufgabe:

(wie 7.1) es werden jedoch mehrere Elemente mit dem selben Indizes addiert.

Hinweis:

Die Variable z stellt die Matrizennummer dar, so daß die Variable M(z,x,y) das Element mit dem Indizes x und y der z-ten Matrize beinhaltet.

Programmlisting:

REM Mehrdimensionale Felder: → Programm 7.2

```
DO
CLS
PRINT "*****"
PRINT "*          Addition beliebig vieler Matrizen          *"
PRINT "*****"
PRINT
PRINT
PRINT "Matrizendimensionierung (Zeilen x Spalten)"
INPUT "Zeilenzahl : ", i
INPUT "Spaltenzahl : ", j
INPUT "Matrizenanzahl : ", n
REDIM M(n, i, j)
PRINT
FOR z = 1 TO n
  PRINT
  PRINT z; ". Matrize"
  PRINT
  FOR x = 1 TO i
    FOR y = 1 TO j
      PRINT "("; x; ", "; y; ")";
      INPUT M(z, x, y)
    NEXT y
  NEXT x
NEXT z
REDIM C(i, j)
FOR x = 1 TO i
  FOR y = 1 TO j
    C(x, y) = 0
    FOR z = 1 TO n
      C(x, y) = C(x, y) + M(z, x, y)
    NEXT z
  NEXT y
NEXT x
CLS
PRINT "*****"
PRINT "*          Addition beliebig vieler Matrizen          *"
PRINT "*****"
PRINT
PRINT
PRINT "Lösungsmatrize"
PRINT
PRINT
FOR x = 1 TO i
  FOR y = 1 TO j
    PRINT C(x, y);
  NEXT y
NEXT x
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabfrage$
LOOP UNTIL tastaturabfrage$ = "n" OR tastaturabfrage$ = "N"
END
```

7.3 Skalarprodukt berechnen

Aufgabe:

Das Skalarprodukt $a \bullet b = a_x \cdot b_x + a_y \cdot b_y + a_z \cdot b_z$ (wie 7.1)

Das Skalarprodukt bildet man, indem man die einzelnen Komponenten mit gleichem Index addiert. Das heißt das Skalarprodukt ist eine Matrizenaddition mit einer einspaltigen Matrix.

Programmlisting:

REM Mehrdimensionale Felder: ➔ Programm 7.3

DO

```
CLS
PRINT "*****"
PRINT "* Berechnung des Skalarproduktes dreidimensionaler Vektoren *"
PRINT "*****"
PRINT
PRINT
REDIM A(3), B(3)
PRINT
PRINT "Vektor 1"
PRINT
FOR i = 1 TO 3
    PRINT "Komponente "; i;
    INPUT "=", A(i)
NEXT i
PRINT
PRINT "Vektor 2"
PRINT
FOR i = 1 TO 3
    PRINT "Komponente "; i;
    INPUT "=", B(i)
NEXT i
s = 0
FOR i = 1 TO 3
    s = s + A(i) * B(i)
NEXT i
PRINT
PRINT
LOCATE 14, 35 '➔ Lokalisiert Zeile 14 und Spalte 35 für nächsten Befehl
PRINT "Skalarprodukt = ", s
LOCATE 24, 5 '➔ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n :", tastaturabfrage$
LOOP UNTIL tastaturabfrage$ = "n" OR tastaturabfrage$ = "N"
END
```

7.4 Multiplikation zweier Matrizen

Aufgabe:

Das Element $C_{i,k}$ der Lösungsmatrix entsteht durch das Skalarprodukt der i-ten Zeile der Matrix **A** und der k-ten Spalte der Matrix **B**. Die Spaltenzahl von **A** muß mit der Zeilenzahl von **B** übereinstimmen.

Hinweis:

Die ordentliche Zuordnung der Variablen ist hier entscheidend. Die Variablen i,j,m und n sind feste (globale) Werte, die am Anfang des Programmes eingegeben werden. Sie geben die Größe der beiden Matrizen an. Die Variablen x,y und z sind Laufvariablen und bei der Berechnung von Bedeutung. X läuft die Zeilen und y die Spalten durch, wobei die FOR-NEXT-Schleife der Variablen z als Skalarprodukt berechnet und in dem jeweiligen Element ablegt.

Programmlisting:

```
REM Mehrdimensionale Felder: → Programm 7.4
DO
  DO
  CLS
  PRINT "*****"
  PRINT "*           Multiplikation zweier Matrizen           *"
  PRINT "*****"
  PRINT
  PRINT
  PRINT
  PRINT "Matrize 1"
  PRINT
  INPUT "Zeilenzahl : ", i
  INPUT "Spaltenzahl : ", j
  PRINT
  PRINT "Matrize 2"
  PRINT
  INPUT "Zeilenzahl : ", k
  INPUT "Spaltenzahl : ", l
  IF j <> k THEN
    PRINT
    COLOR 20 '→ Nächster Befehl wird rot/blinkend ausgegeben
    PRINT "Spaltenanzahl der erste Matrize darf nicht ungleich der Zeilenanzahl"
    PRINT " der zweiten Matrize sein!"
    PRINT
    COLOR 7, 0 '→ Setzt auf ursprüngliche Farben zurück
    PRINT "Betätige die Taste " + "n" + " zur Neueingabe!"
    DO
    LOOP UNTIL INKEY$ = "n"
  END IF
  LOOP UNTIL j = k
  REDIM A(i, j), B(k, l)
  PRINT
  PRINT "Matrize 1"
  PRINT
  FOR x = 1 TO i
    FOR y = 1 TO j
      PRINT "("; x; ","; y; ")";
      INPUT " = ", A(x, y)
    NEXT y
  NEXT x
  PRINT
  PRINT "Matrize 2"
  PRINT
  FOR x = 1 TO i
    FOR y = 1 TO j
      PRINT "("; x; ","; y; ")";
      INPUT " = ", B(x, y)
    NEXT y
  NEXT x
```

```
REDIM C(i, k)
FOR x = 1 TO i
  FOR y = 1 TO k
    C(x, y) = 0
    FOR z = 1 TO j
      C(x, y) = C(x, y) + A(x, z) * B(z, y)
    NEXT z
  NEXT y
NEXT x
CLS
PRINT "*****"
PRINT "*           Multiplikation zweier Matrizen           *"
PRINT "*****"
PRINT
PRINT
PRINT
PRINT
PRINT "Lösungsmatrize : "
PRINT
FOR x = 1 TO i
  FOR y = 1 TO k
    PRINT C(x, y),
  NEXT y
NEXT x
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5
INPUT "Noch mal durchrechnen? j/n : ", tastaturabfrage$
LOOP UNTIL tastaturabfrage$ = "n" OR tastaturabfrage$ = "N"
END
```

7.5 Determinante nach Sarrus bestimmen

Aufgabe:

Den Determinantenwert erhält man, in dem man die drei Hauptdiagonalenprodukte addiert und von der Summe die drei Nebendiagonalen subtrahiert.

Programmlisting:

```
DECLARE SUB Ausgabe ()
REM Mehrdimensionale Felder: → Programm 7.5
DIM SHARED substart, s '→ Variablen allen Subs zugänglich machen

DO
CLS
PRINT "*****"
PRINT "*          Determinate nach Sarrus bestimmen          *"
PRINT "*****"
PRINT
PRINT
PRINT
PRINT
PRINT
REDIM SHARED M(3, 3)
FOR x = 1 TO 3
  FOR y = 1 TO 3
    PRINT "("; x; ", "; y; ")";
    INPUT " = ", M(x, y)
  NEXT y
NEXT x
a = M(1, 1) * M(2, 2) * M(3, 3) + M(1, 2) * M(2, 3) * M(3, 1) + M(1, 3) * M(2, 1) * M(3, 2)
b = -M(1, 3) * M(2, 2) * M(3, 1) - M(1, 2) * M(2, 1) * M(3, 3) - M(1, 1) * M(2, 3) * M(3, 2)
s = a + b
'Ausgabe          '→ man probiere ohne Unterprogramm aus
Ausgabe          '→ Sub Ausgabe starten

IF substart = 0 THEN
  PRINT
  PRINT
  PRINT "Determinate : "; s
END IF
substart = 0 '→ rücksetzen für neuen Programmdurchlauf
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabfrage$
LOOP UNTIL tastaturabfrage$ = "n" OR tastaturabfrage$ = "N"
END

SUB Ausgabe
REDIM M$(3, 3)
substart = 1
groesste$ = STR$(M(1, 1))
FOR x = 1 TO 3
  FOR y = 1 TO 3
    M$(x, y) = STR$(M(x, y))
    IF LEN(M$(x, y)) > LEN(groesste$) THEN '→ bestimmt längste Zeichenkette
      groesste$ = M$(x, y)
    END IF
  NEXT y
NEXT x
FOR x = 1 TO 3
  FOR y = 1 TO 3
    IF LEN(M$(x, y)) < LEN(groesste$) THEN '--> macht alle Strings gleich lang
      M$(x, y) = STRING$(LEN(groesste$) - LEN(M$(x, y)), " ") + M$(x, y)
    END IF
  NEXT y
NEXT x
CLS
PRINT "*****"
PRINT "*          Determinate nach Sarrus bestimmen          *"
PRINT "*****"
PRINT
PRINT
PRINT
PRINT
PRINT
PRINT "+" + STRING$(LEN(groesste$) * 3 + 3, " ") + "+" '→ Fügt Leerzeichen ein
PRINT "i" + M$(1, 1) + " " + M$(1, 2) + " " + M$(1, 3) + " i"
PRINT "i" + M$(2, 1) + " " + M$(2, 2) + " " + M$(2, 3) + " i" = "; s
```

```
PRINT "|" + M$(3, 1) + " " + M$(3, 2) + " " + M$(3, 3) + "|"  
PRINT "+" + STRING$(LEN(groesste$) * 3 + 3, " ") + "+"  
END SUB
```


8. Differentiation von ganzrationalen Funktionen

Aufgabe:

Schreibe ein Programm, das eine ganzrationale Funktion n-ten Grades j-mal differenziert.

Hinweis:

Es werden nur der Grad und die Koeffizienten der Funktion eingelesen.

Programmlisting:

```
DECLARE SUB Ausgabe ()
REM Differentiation von ganzrationalen Funktionen: → Programm 8.1

DIM SHARED i, anz, n '→ Variablen allen Subs zugänglich machen

DO
CLS
PRINT "*****"
PRINT "* Differentiation einer ganzrationalen Funktionen n-ten Grades *"
PRINT "*****"
PRINT
PRINT
PRINT
PRINT
PRINT
INPUT "Grad der einzugebenden Funktion (>1): ", n
INPUT "Wie oft soll sie abgeleitet werden? : ", j
REDIM SHARED K(n)
FOR i = n TO 0 STEP -1
    PRINT "Koeffizient von x^"; i;
    INPUT "= ", K(i)
NEXT i
Ausgabe '→ Sub starten
FOR anz = 1 TO j '→ Schleife für Ableitungen
    FOR i = 0 TO n - 1 '→ Schleife für Koeffizienten
        K(i) = K(i + 1) * (i + 1) '→ Ableitung berechnen
    NEXT i
    IF n = 0 THEN
        K(0) = 0
    END IF
    n = n - 1
    Ausgabe '→ Sub starten
NEXT anz

LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabfrage$
LOOP UNTIL tastaturabfrage$ = "n" OR tastaturabfrage$ = "N"
END

SUB Ausgabe
PRINT
PRINT "f"; anz; " = ";
FOR i = n TO 1 STEP -1
    PRINT K(i); "X^"; i; "+";
NEXT i
PRINT K(0)
END SUB
```

9. Numerische Integration von beliebigen Funktionen

9.1 Trapezformel

Aufgabe:

Gegeben ist eine Funktion, die im Intervall (a,b) die x-Achse nicht schneidet. Schreibe ein Programm, daß mit Hilfe der Trapezformel die Fläche der Funktion im Intervall (a,b) berechnet.

Hinweis:

Die Trapezformel lautet:

$$A_T = h \cdot \left(\frac{(y_0 + y_n) + (y_1 + y_2 + \dots + y_{n-1})}{2} \right)$$

Programmlisting:

REM Numerische Integration von beliebigen Funktionen: → Programm 9.1

```
DEF FNF (x) = 1 / x
DO
CLS
PRINT "*****"
PRINT "*           Numerische Integration - Trapezformel           *"
PRINT "*****"
PRINT
PRINT
PRINT "Trapezformel :"0 + yn) + (y1 + y2) + ... + yn-1 |"
PRINT "          |                                     |"
PRINT "          +                                     +"
PRINT
PRINT "Die Funktion lautet: f(x) = 1/x"
PRINT
INPUT "Eingabe der unteren Grenze: ", a
INPUT "Eingabe der oberen Grenze: ", b
PRINT
INPUT "Anzahl der Teilflächen : ", n
h = (b - a) / n
Summe1 = 1 / 2 * (FNF(a) + FNF(b))
FOR i = 1 TO n - 1
Summe2 = Summe2 + FNF(a + i * h)
NEXT i
F = h * (Summe1 + Summe2)
PRINT
PRINT
PRINT "Die Größe der Fläche ist: "; F
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabfrage$
LOOP UNTIL tastaturabfrage$ = "n" OR tastaturabfrage$ = "N"
END
```

9.2 Simpsonformel

Aufgabe:

Die Aufgabe aus 9.1 ist mit Hilfe der Simpson - Formel zu lösen.

Hinweis:

Die Simpson - Formel lautet:

$$A_S = h \cdot \left(\frac{(y_0 + y_n) + 4(y_1 + y_3 + y_5 + \dots + y_{n-1}) + 2(y_2 + y_4 + \dots + y_{n-2})}{3} \right)$$

Programmlisting:

REM Numerische Integration von beliebigen Funktionen: ➔ Programm 9.2

```
DEF FNF (x) = 1 / x
DO
DO
io = 0
CLS
PRINT "*****"
PRINT "*      Numerische Integration - Simpsonformel      *"
PRINT "*****"
PRINT
PRINT
PRINT
PRINT " Die Funktion lautet: f(x) = 1/x"
PRINT
INPUT "Eingabe der unteren Grenze      : ", a
INPUT "Eingabe der oberen Grenze      : ", b
PRINT
INPUT "Anzahl der Teilflächen (gerade) : ", n
IF n / 2 <> INT(n / 2) THEN
COLOR 20      '➔ Nächster Befehl wird rot/blinkend ausgegeben
PRINT "Die Anzahl der Teilflächen muß gerade sein!"
COLOR 7, 0    '➔ Setzt auf ursprüngliche Farben zurück
PRINT "Bitte Taste betätigen"
io = 1
DO
taste$ = INKEY$      '➔ Setzt Programm erst nach
LOOP UNTIL LEN(taste$) <> 0      ' Betätigung einer Taste fort
END IF
LOOP UNTIL io = 0
h = (b - a) / n
Summe1 = 1 / 2 * (FNF(a) + FNF(b))
Summe2 = 0
Summe3 = 0
FOR i = 1 TO n - 1 STEP 2
Summe2 = Summe2 + FNF(a + i * h)
NEXT i
FOR i = 1 TO n - 2 STEP 2
Summe3 = Summe3 + FNF(a + i * h)
NEXT i
F = h / 3 * (Summe1 + 4 * Summe2 + 2 * Summe3)
PRINT
PRINT
PRINT "Die Größe der Fläche ist: "; F
LOCATE 24, 5      '➔ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n : ", tastaturabfrage$
LOOP UNTIL tastaturabfrage$ = "n" OR tastaturabfrage$ = "N"

END
```

10. Umrechnung von Zahlensystemen

10.1 Umwandlung Dualzahl in Dezimalzahl

Aufgabe:

Eine Zahl soll aus dem Dualen Zahlensystem in das Dezimale Zahlensystem umgewandelt werden.

Programmlisting:

'Umrechnen von Dualzahlen in Dezimalzahlen → 10.1

```
DIM dualstring AS STRING, d AS STRING
DIM dualzahl AS INTEGER, laenge AS INTEGER
CONST falsch = 0
wahr = NOT falsch

DO
  DO
    CLS
    neustart = falsch
    CLS
    PRINT "*****"
    PRINT "*          Umrechnung von Zahlensystemen          *"
    PRINT "*          Dualzahl in Dezimalzahl                    *"
    PRINT "*****"
    PRINT
    PRINT
    PRINT
    INPUT "Eingabe einer Dualzahl: ", dualstring
    dualzahl = 0
    laenge = LEN(dualstring)
    FOR i = 1 TO laenge
      d = MID$(dualstring, laenge - i + 1, 1)
      IF d <> "0" AND d <> "1" THEN
        neustart = wahr
        EXIT FOR
      END IF
      dualzahl = dualzahl + (ASC(d) - 48) * 2 ^ (i - 1)
    NEXT i
    PRINT
    PRINT
    PRINT dualstring; " entspricht "; dualzahl
  LOOP WHILE neustart = wahr
  LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
  INPUT "Noch mal durchrechnen? j/n :", tastaturabrage$
  LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

10.2 Dezimalzahl in Dualzahl

Aufgabe:

Eine Zahl soll aus dem Dezimalen Zahlensystem in das Duale Zahlensystem umgewandelt werden.

Programmlisting:

'Umrechnen von Dezimalzahlen in Dualzahlen 10.2

```
DIM dualstring AS STRING
DIM dezimalzahl AS INTEGER, d AS INTEGER, rest AS INTEGER

DO
CLS
PRINT "*****"
PRINT "*      Umrechnung von Zahlensystemen      *"
PRINT "*      Dezimalzahl in Dualzahl              *"
PRINT "*****"
PRINT
PRINT
PRINT
INPUT "Eingabe einer Dezimalzahl: ", dezimalzahl
PRINT
d = dezimalzahl
dualstring = ""
DO
rest = (d / 2 - INT(d / 2)) * 2
dualstring = CHR$(ASC("0") + rest) + dualstring
PRINT d, rest
d = INT(d / 2)
LOOP WHILE d > 0
PRINT
PRINT dezimalzahl; " entspricht "; dualstring
LOCATE 24, 5 '➔ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n :", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

10.3 Dezimalzahl in Hexadezimalzahl

Aufgabe:

Eine Zahl soll aus dem Dezimalen Zahlensystem in das Hexadezimale Zahlensystem umgewandelt werden.

Programmlisting:

'Umwandlung von Dezimal in Hexadezimal 10.3

```
DO
CLS
PRINT "*****"
PRINT "*      Umrechnung von Zahlensystemen      *"
PRINT "*      Dezimalzahl in Hexadezimalzahl      *"
PRINT "*****"
PRINT
PRINT
PRINT
INPUT "Geben Sie eine Dezimalzahl ein! ", x
PRINT
PRINT
PRINT "Ihre hexadezimale Darstellung lautet: "; HEX$(x)
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n :", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

10.4 Hexadezimalzahl in Dezimalzahl

Aufgabe:

Eine Zahl soll aus dem Hexadezimalen Zahlensystem in das Dezimale Zahlensystem umgewandelt werden.

Programmlisting:

'Umwandlung von Hexadezimal in Dezimal → 10.4

'Umwandlung funktioniert nur im Integer-Bereich

```
DO
CLS
PRINT "*****"
PRINT "*      Umrechnung von Zahlensystemen      *"
PRINT "*      Hexadezimalzahl in Dezimalzahl      *"
PRINT "*****"
PRINT
PRINT
PRINT
INPUT "Geben Sie eine Hex-Zahl ein! ", x$
PRINT
PRINT "Ihre dezimale Darstellung lautet: "; VAL("&H" + x$)
LOCATE 24, 5 '→ Lokalisiert Zeile 24 und Spalte 5 für nächsten Befehl
INPUT "Noch mal durchrechnen? j/n :", tastaturabrage$
LOOP UNTIL tastaturabrage$ = "n" OR tastaturabrage$ = "N"
END
```

----- Ende des Dokuments -----