

```

DECLARE SUB Neu (DName$)
DECLARE SUB ZeigeText (XPos%, YPos%, Text$)
DECLARE SUB Test (Teil%)
DECLARE SUB ZeigeMenue (TeilNummer%)
DECLARE SUB ZeigeBand (BPos%)
DECLARE SUB Rahmen (XS%, YS%, XE%, YE%, RT%)
DECLARE SUB LoescheBereich ( )
DECLARE SUB Optionen (BE$, BF$, PE$, PF$, Modus%)
DECLARE SUB Ende (Dummy%)
DECLARE SUB Starten (Trace$, BStart$)
DECLARE FUNCTION Existiert! (Name$)
DECLARE SUB ZeigeProgramm (PZeile%)
DECLARE SUB EditZeile (XPos%, YPos%, Anzahl%, DLaenge%, Inh$, Art%)
DECLARE SUB ZeigeMeldung (AZus$, BPos%, Rich$, Art%)
DECLARE SUB LiesZeichen (In$)
DECLARE SUB ZeigeTitel ( )
DECLARE SUB ZeigeVariablen (AZus$, NZus$, AZei$, NZei$, Rich$, BPos%)
DECLARE SUB Voreinstellung (Trace$, TBName$, TPName$, BStart$, Modus%)
DECLARE SUB LadeBand (TBName$)
DECLARE SUB ZeigeHilfe (Seite%)
DECLARE SUB EditBand (BE$, BF$, TBName$)
DECLARE SUB SpeichereBand (TBName$)
DECLARE SUB SpeichereProgramm (TPName$)
DECLARE SUB LadeProgramm (TPName$)
DECLARE SUB EditProgramm (PE$, PF$, TPName$)
DECLARE SUB Farbe (VCol%, HCol%)

REM * ----- *
REM *      Turing-Interpreter 2.10      *
REM *      ===== *
REM * Programmiert von : Sebastian Mory *
REM * Fertiggestellt am : 10.07.1991   *
REM * * *
REM * Alle Rechte vorbehalten *
REM * 1991 bei Mocom (C) *
REM * ----- *

SCREEN 0, 0, 0
Farbe 0, 0
CLS

' 1. Globale Variablen definieren.
' -----

DIM SHARED BMax%
DIM SHARED PMax%
DIM SHARED LNr$
LNr$ = "4B616C6D62616368"

BMax% = 500
PMax% = 200

DIM SHARED Band(1 TO BMax%) AS STRING * 1 ' Bandl,nge (500 Zeichen).
DIM SHARED Prog(1 TO PMax%) AS STRING * 13 ' Prg.-l,nge (200 Zeilen).

DIM SHARED VCol%, HCol%, HSeite% ' Farben und Nummer der Hilfeseite.

HSeite% = 1

ON KEY(1) GOSUB Helfen ' Hilfeseiten anzeigen, wenn F1 gedrückt worden ist.

KEY(1) ON

' ON ERROR GOTO Fehler ' Echte Funktionsfehler werden abgefangen.

```

```
' 2. Konfigurationsdateien auf Existenz prüfen.
```

```
' -----
```

```
ZeigeMeldung AZus$, BPos%, Rich$, 3
```

```
HSeite% = 1
```

```
SLEEP 5
```

```
DateiEx% = 1
```

```
DateiEx% = Existiert("Turing1.Cfg")
```

```
DateiEx% = Existiert("Turing2.Cfg")
```

```
IF DateiEx% = 0 THEN
```

```
  HSeite% = 1
```

```
  GOSUB Helfen
```

```
END IF
```

```
DateiEx% = 1
```

```
' Nun werden die Daten aus den Konfigurationsdateien gelesen.
```

```
' Wenn noch keine Daten eingegeben wurden, werden stattdessen
```

```
' die Daten aus der Prozedur Test verwendet.
```

```
Voreinstellung Trace$, TBName$, TPName$, BStart$, 0
```

```
HSeite% = 2
```

```
IF LEN(TBName$) = 0 THEN
```

```
  Test 1
```

```
ELSE
```

```
  DateiEx% = Existiert(TBName$)
```

```
  IF DateiEx% = 0 THEN
```

```
    ZeigeMeldung AZus$, BPos%, Rich$, 4
```

```
    Test 1
```

```
  ELSE
```

```
    LadeBand TBName$
```

```
  END IF
```

```
END IF
```

```
IF LEN(TPName$) = 0 THEN
```

```
  Test 2
```

```
ELSE
```

```
  DateiEx% = Existiert(TPName$)
```

```
  IF DateiEx% = 0 THEN
```

```
    ZeigeMeldung AZus$, BPos%, Rich$, 5
```

```
    Test 2
```

```
  ELSE
```

```
    LadeProgramm TPName$
```

```
  END IF
```

```
END IF
```

```
' Falls keine externen Band- bzw. Programmdateien verwendet werden,
```

```
' werden den internen Dateien Namen zugewiesen, um sie von den evtl.
```

```
' gewünschten externen Editoren trotzdem editieren zu lassen.
```

```
DateiEx% = 1
```

```
Optionen BE$, BF$, PE$, PF$, 0
```

```
HSeite% = 3
```

```
IF BE$ = CHR$(0) THEN BE$ = ""
```

```

IF BF$ = CHR$(0) THEN BF$ = ""
IF PE$ = CHR$(0) THEN PE$ = ""
IF PF$ = CHR$(0) THEN PF$ = ""

IF LEN(BE$) = 0 THEN
ELSE
  DateiEx% = Existiert(BF$ + "\" + BE$)
  IF DateiEx% = 0 THEN
    ZeigeMeldung AZus$, BPos%, Rich$, 6
  END IF
  DateiEx% = 1
END IF

IF LEN(PE$) = 0 THEN
ELSE
  DateiEx% = Existiert(PF$ + "\" + PE$)
  IF DateiEx% = 0 THEN
    ZeigeMeldung AZus$, BPos%, Rich$, 7
  END IF
  DateiEx% = 1
END IF

' Wenn diese Daten nicht eingestellt wurden, werden die internen
' Editoren verwendet.

' Es werden noch ben"tigte Variablen zugewiesen.

IF Trace$ = "" OR Trace$ = CHR$(0) THEN Trace$ = ""
IF TBName$ = "" OR TBName$ = CHR$(0) THEN TBName$ = "TestBand.Tur"
IF TPName$ = "" OR TPName$ = CHR$(0) THEN TPName$ = "TestProg.Tur"
IF BStart$ = "" OR BStart$ = CHR$(0) THEN BStart$ = "99"

' Ein bi"chen Optik vorab.
' -----

HSeite% = 4
ZeigeMenue 0

' Programmschleife.
' -----

DO ' Programmschleife.

  DO ' Men"schleife.

    HSeite% = 4
    Ein$ = In$
    Farbe 14, 5

    SELECT CASE In$ ' Markierung der Men"punkte.

      CASE "B", "b"
        LOCATE 2, 4, 0
        PRINT "Band editieren"
        HSeite% = 5

      CASE "P", "p"
        LOCATE 2, 20, 0
        PRINT "Programm editieren"
        HSeite% = 6

      CASE "S", "s"
        LOCATE 2, 40, 0
        PRINT "Starten"

```

```

        HSeite% = 7

    CASE "V", "v"
        LOCATE 2, 49, 0
        PRINT "Voreinstellung"
        HSeite% = 8

    CASE "O", "o"
        LOCATE 2, 65, 0
        PRINT "Optionen"
        HSeite% = 9

    CASE "E", "e"
        LOCATE 2, 75, 0
        PRINT "Ende"
        HSeite% = 10

    CASE ELSE

END SELECT

LiesZeichen In$

IF In$ <> CHR$(13) THEN
    ZeigeMenue 1
    ZeigeMenue 2
END IF

LOOP UNTIL In$ = CHR$(13) ' Men◆schleife.

SELECT CASE Ein$ ' Auswahl der Men◆punkte.

    CASE "B", "b"
        EditBand BE$, BF$, TBName$

    CASE "P", "p"
        EditProgramm PE$, PF$, TPName$

    CASE "S", "s"
        Starten Trace$, BStart$

    CASE "V", "v"
        Voreinstellung Trace$, TBName$, TPName$, BStart$, 1

        IF TBName$ = "" THEN TBName$ = "TestBand.Tur"
        IF TPName$ = "" THEN TPName$ = "TestProg.Tur"
        IF LEN(BStart$) = 0 THEN BStart$ = "99"

    CASE "O", "o"
        Optionen BE$, BF$, PE$, PF$, 1

    CASE "E", "e"
        Ende Dummy%
        IF Dummy% = 1 THEN EXIT DO

    CASE ELSE

END SELECT

ZeigeMenue 1
ZeigeMenue 2

LOOP ' Programmschleife.

```

KEY(1) OFF

ZeigeTitel
Farbe 7, 0

CLS

END

Helfen:

KEY(1) OFF

' Hier wird die Cursorposition gespeichert.
' Die Farben sind in VCol% und HCol% gespeichert.

CXPos% = CSRLIN
CYPos% = POS(0)

ZeigeHilfe Seite%

' Nun werden die Farben und die Cursorposition wiederhergestellt.

LOCATE CXPos%, CYPos%, 0
Farbe VCol%, HCol%

KEY(1) ON

RETURN

Fehler:

' Echte Funktionsfehler werden abgefangen.

CXPos% = CSRLIN
CYPos% = POS(0)
SysFehler% = ERR

ZeigeMeldung AZus\$, BPos%, Rich\$, SysFehler%

IF SysFehler% = 0 THEN ' Der Fehler ist nicht kritisch.
CLEAR ' Alle Variablen werden zurückgesetzt,
' RUN ' und das Programm wird erneut gestartet.
END IF

' Evtl. kann dann weitergemacht werden.
' Momentan aber wird generell angehalten.

END

SUB EditBand (BE\$, BF\$, TBName\$)

' Diese Prozedur startet den gewünschten Editor zum Editieren des Bandes,
' oder, wenn keiner angegeben ist, den internen.

Farbe 12, 0

Rahmen 9, 1, 11, 80, 2

LOCATE 10, 3, 0
PRINT " Bitte gedulden Sie sich einen Moment ..."

```

' Wenn das Testband gewünscht wird, muß es
' aus der Prozedur Test gelesen werden.

IF TBName$ = "TestBand.Tur" THEN Test 2

' Das Band wird als Datei gespeichert.

SpeichereBand TBName$

' Jetzt wird entschieden, welcher Editor verwendet werden soll.

IF BE$ = "" THEN

    ' Internen Editor verwenden.
    ' Bevor das Band editiert werden kann, muß es in einen String
    ' der Länge BMax% umgewandelt werden.

    EdBand$ = ""
    FOR I% = 1 TO BMax%
        EdBand$ = EdBand$ + Band(I%)
    NEXT I%

    LOCATE 14, 1, 0
    Farbe 0, 3
    PRINT " Cursortasten zum Bewegen des Cursors"
    PRINT " [Ctrl.] + Cursortasten zum Springen zur nächsten Tab-Position"
    PRINT " [#] als leeres Element verwenden."
    PRINT " [0] als Null - Element verwenden."
    PRINT " [1] als Eins - Element verwenden."
    PRINT
    PRINT " Andere Elemente (Zahlen, Buchstaben, andere Zeichen) sind zwar
erlaubt,"
    PRINT " erfordern allerdings auch angepasste Turing-Programme - sie sollten
daher"
    PRINT " möglichst nicht verwendet werden."

    Farbe 14, 0
    EditZeile 10, 4, 73, BMax%, EdBand$, 2

    FOR I% = 1 TO BMax%
        Band(I%) = MID$(EdBand$, I%, 1)
    NEXT I%

ELSE

    ' Externen Editor verwenden.

    SHELL "Cd >Turing0.cfg"
    ' Aktuelles Laufwerk und Pfad zwischenspeichern.

    KEY(1) OFF

    OPEN "R", #1, "Turing0.cfg", 1
    FIELD #1, 1 AS Z$

    I% = 1
    Ablage$ = ""

    DO WHILE NOT EOF(1)
        GET #1, I%
        Ablage$ = Ablage$ + Z$
        I% = I% + 1
    LOOP

```

```

CLOSE #1

KEY(1) ON

' Nun wird je nach Aufbau des Pfads zuerst das Laufwerk gewechselt oder
' direkt der Pfad aufgerufen.

IF MID$(BF$, 2, 1) = ":" THEN
  SHELL LEFT$(BF$, 2)
  SHELL "Cd " + RIGHT$(BF$, LEN(BF$) - 2)
ELSE
  SHELL "Cd " + BF$
END IF
' Nun wird der Editor aufgerufen.
Farbe 7, 0
CLS
SHELL BE$ + " " + LEFT$(Ablage$, LEN(Ablage$) - 3) + "\" + TBName$

' Nun muß wieder in das aktuelle Laufwerk und das aktuelle
' Verzeichnis gewechselt werden.

SHELL LEFT$(Ablage$, 2)
SHELL "Cd " + Ablage$

' Nun muß das Band wieder geladen werden.

LadeBand TBName$

' Falls durch den Fremdeditor Returnzeichen eingefügt wurden, werden diese
' durch Leerfelder ersetzt.

' FOR I% = 1 TO BMax%
'   T$ = MID$(EdBand$, I%, 1)
'   IF T$ = CHR$(13) THEN T$ = "#"
'   Band(I%) = MID$(EdBand$, I%, 1)
'   NEXT I%

ZeigeMenue 0

END IF

LoescheBereich

END SUB

SUB EditProgramm (PE$, PF$, TPName$)

' Diese Prozedur startet den gewünschten Editor zum Editieren des
' Turing-Programmes, oder, wenn keiner angegeben ist, den internen.

' Wenn das Testprogramm gewünscht wird, muß es
' aus derProzedur Test gelesen werden.

IF TPName$ = "TestProg.Tur" THEN Test 1

' Das Turing-Programm wird als Datei gespeichert.

SpeichereProgramm TPName$

' Jetzt wird entschieden, welcher Editor verwendet werden soll.

IF PE$ = "" THEN

  ' Internen Editor verwenden.

```



```
LOCATE 7 + I% - SZeile%, 3, 0
PRINT SPACE$(13)
```

```
LiesZeichen In$
```

```
SELECT CASE RIGHT$(In$, 1)

    CASE CHR$(27)
        EXIT DO

    CASE "N", "n"
        IF Prog(PMax%) <> "" THEN
            FOR I% = PMax% TO AZeile% STEP -1
                IF I% > 1 THEN Prog(I%) = Prog(I% - 1)
            NEXT I%
            Prog(AZeile%) = SPACE$(13)
        END IF

    CASE "L", "l"
        FOR I% = AZeile% TO EZeile% + 1
            Prog(I%) = Prog(I% + 1)
        NEXT I%

    CASE CHR$(72)
        IF AZeile% > 1 THEN AZeile% = AZeile% - 1

    CASE CHR$(80)
        IF AZeile% < EZeile% THEN AZeile% = AZeile% + 1

    CASE ELSE
        EditZeile 21, 3, 13, 13, Prog(AZeile%), 3
        LOCATE 21, 3, 0
        PRINT SPACE$(13)
```

```
END SELECT
LOOP
```

```
ELSE
```

```
' Externen Editor verwenden.
```

```
SHELL "Cd >Turing0.cfg"
```

```
' Aktuelles Laufwerk und Pfad zwischenspeichern.
```

```
KEY(1) OFF
```

```
OPEN "R", #1, "Turing0.cfg", 1
FIELD #1, 1 AS Z$
```

```
I% = 1
Ablage$ = ""
```

```
DO WHILE NOT EOF(1)
    GET #1, I%
    Ablage$ = Ablage$ + Z$
    I% = I% + 1
LOOP
```

```
CLOSE #1
```

```
KEY(1) ON
```

```
' Nun wird je nach Aufbau des Pfads zuerst das Laufwerk gewechselt oder
' direkt der Pfad aufgerufen.
```

```

IF MID$(PF$, 2, 1) = ":" THEN
  SHELL LEFT$(PF$, 2)
  SHELL "Cd " + RIGHT$(PF$, LEN(PF$) - 2)
ELSE
  SHELL "Cd " + PF$
END IF
' Nun wird der Editor aufgerufen.
Farbe 7, 0
CLS
SHELL PE$ + " " + LEFT$(Ablage$, LEN(Ablage$) - 3) + "\" + TPName$

' Nun muá wieder in das aktuelle Laufwerk und das aktuelle
' Verzeichnis gewechselt werden.

SHELL LEFT$(Ablage$, 2)
SHELL "Cd " + Ablage$

' Nun muá das Turing-Programm wieder geladen werden.

LadeProgramm TPName$

ZeigeMenue 0

END IF

LoescheBereich

END SUB

SUB EditZeile (XPos%, YPos%, Anzahl%, DLaenge%, Inh$, Art%)

' In dieser Prozedur wird eine  $\diamond$ bergebene Textzeile editiert.

CPos% = 1 ' Position, an der der Cursor steht.
DPos% = 1 ' Position, von der ab maximal Anzahl% Zeichen dargestellt werden.

Ablage$ = Inh$

Bereich$ = "1234567890"

IF Art% > 0 THEN
  Bereich$ = Bereich$ + "ABCDEFGHIJKLMNOPQRSTUVWXYZŽš"
  Bereich$ = Bereich$ + "abcdefghijklmnopqrstuvwxy,,," $\diamond$ á"
  Bereich$ = Bereich$ + " .:_\/[ ]!$%&()"
END IF

IF Art% > 1 THEN
  Bereich$ = Bereich$ + "=?'`@+*~#;,<>|#"
END IF

IF Art% = 2 THEN
  LOCATE 12, 2, 0
  PRINT " Bandposition: 1 "
END IF

DO

  LOCATE XPos%, YPos%, 0

  PRINT MID$(Inh$, DPos%, Anzahl%); " "

```

```
LOCATE XPos%, YPos% + CPos% - DPos%, 1, 6, 7
```

```
LiesZeichen In$
```

```
SELECT CASE RIGHT$(In$, 1)
```

```
  CASE CHR$(8)
```

```
    IF LEN(Inh$) > 0 THEN
```

```
      Dummy% = CPos% - 2
```

```
      IF Dummy% < 0 THEN Dummy% = 0
```

```
      Inh$ = MID$(Inh$, 1, Dummy%) + MID$(Inh$, CPos%, LEN(Inh$))
```

```
      IF CPos% > 1 THEN CPos% = CPos% - 1
```

```
      IF Art% = 2 THEN Inh$ = Inh$ + "#"
```

```
      IF Art% = 3 THEN Inh$ = Inh$ + " "
```

```
    ELSE
```

```
    END IF
```

```
  CASE CHR$(27)
```

```
    Inh$ = Ablage$
```

```
    EXIT DO
```

```
  CASE CHR$(13)
```

```
    EXIT DO
```

```
  CASE CHR$(75)
```

```
    IF LEFT$(In$, 1) = CHR$(0) THEN
```

```
      IF CPos% > 1 THEN CPos% = CPos% - 1
```

```
    END IF
```

```
  CASE CHR$(77)
```

```
    IF LEFT$(In$, 1) = CHR$(0) THEN
```

```
      IF CPos% < LEN(Inh$) + 1 THEN CPos% = CPos% + 1
```

```
    END IF
```

```
  CASE CHR$(115)
```

```
    IF LEFT$(In$, 1) = CHR$(0) THEN
```

```
      IF CPos% > 10 THEN
```

```
        CPos% = CPos% - 10
```

```
      ELSE
```

```
        CPos% = 1
```

```
      END IF
```

```
    END IF
```

```
  CASE CHR$(116)
```

```
    IF LEFT$(In$, 1) = CHR$(0) THEN
```

```
      IF CPos% < LEN(Inh$) - 9 THEN
```

```
        CPos% = CPos% + 10
```

```
      ELSE
```

```
        CPos% = LEN(Inh$)
```

```
      END IF
```

```
    END IF
```

```
  CASE ELSE
```

```
END SELECT
```

```
IF INSTR(Bereich$, In$) <> 0 THEN
```

```
  IF Art% = 0 OR Art% = 1 THEN
```

```
    ' Normaler Text wird editiert.
```

```
    IF LEN(Inh$) < DLaenge% THEN
```

```

        Inh$ = MID$(Inh$, 1, CPos% - 1) + In$ + MID$(Inh$, CPos%, LEN(Inh$))
        CPos% = CPos% + 1
    END IF

ELSE
    ' Das Band oder das Programm wird editiert.
    Inh$ = MID$(Inh$, 1, CPos% - 1) + In$ + MID$(Inh$, CPos% + 1, LEN(Inh$))
    IF CPos% < DLaenge% THEN CPos% = CPos% + 1

    END IF

END IF

DO WHILE CPos% - DPos% > Anzahl%
    DPos% = DPos% + 1
LOOP

DO WHILE CPos% < DPos%
    DPos% = DPos% - 1
LOOP

    'IF CPos% - DPos% > Anzahl% THEN DPos% = DPos% + 1
    'IF CPos% < DPos% THEN DPos% = DPos% - 1

IF Art% = 2 THEN
    LOCATE 12, 2, 0
    PRINT USING " Bandposition: ### "; CPos%
END IF

LOOP

END SUB

SUB Ende (Dummy%)

    ' Diese Prozedur zeigt das Ende-Fenster und wartet auf eine Eingabe.

Farbe 12, 0

KEY(1) OFF

FOR I% = 2 TO 11
    Rahmen 12 - (INT(I% / 3)), 37 - I%, 12 + (INT(I% / 3)), 43 + I%, 0
    J% = I% - 1
    Rahmen 12 - (INT(J% / 3)), 37 - J%, 12 + (INT(J% / 3)), 43 + J%, 2
NEXT I%

LOCATE 12, 29, 0
Farbe 14, 0

PRINT "Programm beenden (j/n)"

LOCATE 25, 1, 0
Farbe 7, 1
PRINT " Tasten [J] [N] f r Programm beenden bzw. nicht beenden
";

KEY(1) ON

LiesZeichen In$

IF In$ = "J" OR In$ = "j" THEN Dummy% = 1

```

LoescheBereich

END SUB

FUNCTION Existiert (Name\$)

' Diese Prozedur "ffnet die Datei, die durch Name\$ bezeichnet ist,
' und liest 3 Zeichen. Wenn dies gelingt, wird 1 zurckgegeben,
' wenn vorher EOF erreicht ist, wird Null zurckgegeben und die
' ge"ffnete Datei wird wieder entfernt.
' Damit ist die Existenz der gewnschten Datei belegt oder widerlegt, was
' eine Fehlermeldung hervorruft.

DaEx% = 1

KEY(1) OFF

OPEN "R", #1, Name\$, 1
FIELD #1, 1 AS Z\$

FOR I% = 1 TO 3
 IF EOF(1) THEN
 DaEx% = 0
 EXIT FOR
 END IF
 GET #1, I%
NEXT I%

Existiert = DaEx%

' Wenn die Dateien Turing1.Cfg oder Turing2.Cfg nicht existieren, werden
' sie erzeugt.

IF DaEx% = 0 AND (Name\$ = "Turing1.Cfg" OR Name\$ = "Turing2.Cfg") THEN
 CLOSE #1
 Neu Name\$
 EXIT FUNCTION
END IF

CLOSE #1

IF DaEx% = 0 THEN KILL Name\$

KEY(1) ON

END FUNCTION

SUB Farbe (VCol%, HCol%)

' In dieser Prozedur werden die bergebenen Werte in die globalen Variablen
' VCol% und HCol% gespeichert und die Farbe wird neu gesetzt.

COLOR VCol%, HCol%

END SUB

SUB LadeBand (TBName\$)

' In dieser Prozedur wird die angegebene Band-Datei in das Band geladen.

KEY(1) OFF

OPEN "R", #1, TBName\$, 1
FIELD #1, 1 AS T\$

```

BPos% = 1

DO WHILE NOT EOF(1)

    GET #1
    IF BPos% < BMax% + 1 THEN
        IF ASC(T$) > 32 THEN
            Band(BPos%) = T$
        ELSE
            Band(BPos%) = "#"
        END IF
        BPos% = BPos% + 1
    END IF
LOOP

CLOSE #1

KEY(1) ON

END SUB

SUB LadeProgramm (TPName$)

' In dieser Prozedur wird das angegebene Turing-Programm geladen.

KEY(1) OFF

OPEN "R", #1, TPName$, 15
FIELD #1, 15 AS T$
PZeile% = 1

DO WHILE NOT EOF(1)

    GET #1
    IF PZeile% < PMax% + 1 THEN
        Abl$ = ""
        FOR I% = 1 TO 13
            IF ASC(MID$(T$, I%, 1)) > 31 THEN
                Abl$ = Abl$ + MID$(T$, I%, 1)
            ELSE
                Prog(PZeile%) = STRING$(13, 0)
            END IF
        NEXT I%
        IF Prog(PZeile%) <> STRING$(13, 0) THEN Prog(PZeile%) = Abl$
        PZeile% = PZeile% + 1
    END IF
LOOP

CLOSE #1

KEY(1) ON

END SUB

SUB LiesZeichen (In$)

' Diese Prozedur wartet auf eine Eingabe von der Tastatur.

DO
    In$ = INKEY$
LOOP WHILE In$ = ""

END SUB

```

SUB LoescheBereich

' Diese Prozedur l"scht den Bildschirm unterhalb der Men"zeile.

KEY(1) OFF

Farbe 0, 3

VIEW PRINT 4 TO 25

CLS

VIEW PRINT

KEY(1) ON

END SUB

SUB Neu (DName\$)

' Diese Prozedur "leert" die Konfigurationsdateien, falls unzul"ssige
' Eintragungen vorhanden sind.

KEY(1) OFF

OPEN "R", #1, DName\$, 1

FIELD #1, 1 AS Z\$

Cfg\$ = "####"

FOR I% = 1 TO 4

 RSET Z\$ = MID\$(Cfg\$, I%, 1)

 PUT #1, I%

NEXT I%

CLOSE #1

KEY(1) ON

END SUB

SUB Optionen (BE\$, BF\$, PE\$, PF\$, Modus%)

' In dieser Prozedur werden der Name des Band-Editors mit seinem Suchpfad und
' der Name des Turing-Programm-Editors mit dessen Suchpfad geladen.
' Dann k"nnen diese Daten - je nach Modus - editiert werden.

' Zuerst werden die gespeicherten Werte geladen und in die entsprechenden
' Variablen eingeordnet.

KEY(1) OFF

OPEN "R", #1, "Turing1.Cfg", 1

FIELD #1, 1 AS Cfgt\$

I% = 1

J% = 1

BE\$ = ""

BF\$ = ""

PE\$ = ""

PF\$ = ""

DO UNTIL EOF(1)

 GET #1, I%

 IF Cfgt\$ = "#" THEN

 J% = J% + 1

```

        Cf$ = ""
    ELSE
        Cf$ = Cfgt$
    END IF

    SELECT CASE J%
        CASE 1
            IF LEN(BE$) < 31 THEN BE$ = BE$ + Cf$

        CASE 2
            IF LEN(BF$) < 61 THEN BF$ = BF$ + Cf$

        CASE 3
            IF LEN(PE$) < 31 THEN PE$ = PE$ + Cf$

        CASE 4
            IF LEN(PF$) < 61 THEN PF$ = PF$ + Cf$

        CASE ELSE
    END SELECT

    I% = I% + 1

LOOP

CLOSE #1

KEY(1) ON

' Hier wird die Prozedur verlassen, wenn nichts editiert werden soll.

IF Modus% = 0 THEN EXIT SUB

' Sonst wird das Fenster ge"ffnet und die Daten werden angezeigt.

KEY(1) OFF

Farbe 12, 0

FOR I% = 2 TO 16
    Rahmen 11 - (INT(I% / 3)), 36 - I%, 9 + (INT(I% / 3)), 44 + I%, 0
    J% = I% - 1
    Rahmen 11 - (INT(J% / 3)), 36 - J%, 9 + (INT(J% / 3)), 44 + J%, 2
NEXT I%

KEY(1) ON

Farbe 14, 0
LOCATE 8, 23, 0
PRINT "Band-Editor      : "; LEFT$(BE$, 16); " "
LOCATE 9, 23, 0
PRINT "Ba.-Verzeichnis : "; LEFT$(BF$, 16); " "
LOCATE 11, 23, 0
PRINT "Programm-Editor : "; LEFT$(PE$, 16); " "
LOCATE 12, 23, 0
PRINT "Pr.-Verzeichnis : "; LEFT$(PF$, 16); " "

Farbe 9, 0
LOCATE 8, 23, 0
PRINT "B"
LOCATE 9, 24, 0
PRINT "a"
LOCATE 11, 23, 0

```



```

PRINT "P"
LOCATE 12, 24, 0
PRINT "r"

Farbe 7, 1
LOCATE 25, 1, 0
PRINT " Tasten [B] [A] [P] [R] f r die Men punkte      [Esc] f r Abbruch
[F1] Hilfe ";

' Nun k nnen Sie editiert werden.

DO
  LiesZeichen In$

  SELECT CASE In$
    CASE "B", "b"
      Farbe 7, 1
      EditZeile 8, 41, 16, 30, BE$, 1
      LOCATE 8, 41, 0
      Farbe 14, 0
      PRINT LEFT$(BE$, 16); " ";
      IF LEN(BE$) < 17 THEN PRINT SPACE$(17 - LEN(BE$))
    CASE "A", "a"
      Farbe 7, 1
      EditZeile 9, 41, 16, 60, BF$, 1
      'IF RIGHT$(BF$, 1) <> "\" AND LEN(BF$) > 0 THEN BF$ = BF$ + "\"
      LOCATE 9, 41, 0
      Farbe 14, 0
      PRINT LEFT$(BF$, 16); " ";
      IF LEN(BF$) < 17 THEN PRINT SPACE$(17 - LEN(BF$))
    CASE "P", "p"
      Farbe 7, 1
      EditZeile 11, 41, 16, 30, PE$, 1
      LOCATE 11, 41, 0
      Farbe 14, 0
      PRINT LEFT$(PE$, 16); " ";
      IF LEN(PE$) < 17 THEN PRINT SPACE$(17 - LEN(PE$))
    CASE "R", "r"
      Farbe 7, 1
      EditZeile 12, 41, 16, 60, PF$, 1
      'IF RIGHT$(PF$, 1) <> "\" AND LEN(PF$) > 0 THEN PF$ = PF$ + "\"
      LOCATE 12, 41, 0
      Farbe 14, 0
      PRINT LEFT$(PF$, 16); " ";
      IF LEN(PF$) < 17 THEN PRINT SPACE$(17 - LEN(PF$))
    CASE CHR$(27)
      EXIT DO
    CASE ELSE
  END SELECT
LOOP

' Nun werden die aktuellen Werte gespeichert.

KEY(1) OFF

OPEN "R", #1, "Turing1.Cfg", 1
FIELD #1, 1 AS Cfgt$

Cfg$ = BE$ + "#" + BF$ + "#" + PE$ + "#" + PF$ + "#"

FOR I% = 1 TO LEN(Cfg$)

  RSET Cfgt$ = MID$(Cfg$, I%, 1)

```

```

    PUT #1, I%
NEXT I%
CLOSE #1
KEY(1) ON
LoescheBereich
END SUB

SUB Rahmen (XS%, YS%, XE%, YE%, RT%)
' Diese Prozedur zeichnet einen leeren, einfachen oder doppelten Rahmen.

SELECT CASE RT%
  CASE 0
    r$ = "      "
  CASE 1
    r$ = "Ú¿ÀÛÄ³"
  CASE 2
    r$ = "É»È¼Í°"
  CASE ELSE
    r$ = "FFFFFF" ' andere Rahmentypen werden hier nicht gebraucht.
END SELECT

OR$ = MID$(r$, 1, 1) + STRING$(YE% - YS% - 1, MID$(r$, 5, 1)) + MID$(r$, 2, 1)
UR$ = MID$(r$, 3, 1) + STRING$(YE% - YS% - 1, MID$(r$, 5, 1)) + MID$(r$, 4, 1)
MR$ = RIGHT$(r$, 1) + SPACE$(YE% - YS% - 1) + RIGHT$(r$, 1)

LOCATE XS%, YS%, 0
PRINT OR$
FOR I% = XS% + 1 TO XE% - 1
  LOCATE I%, YS%, 0
  PRINT MR$
NEXT I%
LOCATE XE%, YS%, 0
PRINT UR$;

END SUB

SUB SpeichereBand (TBName$)
' In dieser Prozedur wird das Turing-Band unter dem angegebenen Namen
' gespeichert.

KEY(1) OFF

OPEN "R", #1, TBName$, 1
FIELD #1, 1 AS T$

FOR I% = 1 TO BMax%
  RSET T$ = Band(I%)
  PUT #1, I%
NEXT I%

CLOSE #1

KEY(1) ON

END SUB

SUB SpeichereProgramm (TPName$)

```

' In dieser Prozedur wird das Turing-Programm unter dem angegebenen Namen
' gespeichert.

KEY(1) OFF

OPEN "R", #1, TPName\$, 15
FIELD #1, 15 AS T\$

FOR I% = 1 TO PMax%
 RSET T\$ = Prog(I%) + CHR\$(13) + CHR\$(10)
 PUT #1, I%
NEXT I%

CLOSE #1

KEY(1) ON

END SUB

SUB Starten (Trace\$, BStart\$)

' Diese Prozedur fñhrt das eigentliche Turing-Programm aus.

Farbe 9, 0

Rahmen 5, 2, 7, 78, 1
LOCATE 5, 3, 0
PRINT " Band "

LOCATE 7, 37, 0
PRINT "¿ ÜÜÜ Ú"
LOCATE 8, 37, 0
PRINT "ÄÄÄÄÄÄÜ"

IF Trace\$ <> "2" THEN
 LOCATE 8, 2, 0
 PRINT " Bandposition: "
END IF

LOCATE 25, 1, 0
Farbe 7, 1

SELECT CASE Trace\$

 CASE "0"
 PRINT " [Esc] fñr Abbruch Turing-Programm lñuft selbstñndig ab."
";

 CASE "1"
 PRINT " [Esc] fñr Abbruch Nñchster Programmschritt durch
beliebigen Tastendruck ";
 Farbe 9, 0
 Rahmen 12, 61, 23, 79, 1
 LOCATE 12, 64, 0
 PRINT " Programm "

 CASE "2"
 PRINT " [Esc] fñr Abbruch Nñchster Programmschritt durch
beliebigen Tastendruck ";
 Farbe 9, 0
 Rahmen 12, 2, 23, 30, 1
 LOCATE 12, 4, 0
 PRINT " Variablen "

```

Rahmen 12, 61, 23, 79, 1
LOCATE 12, 64, 0
PRINT " Programm "

```

```

CASE ELSE

```

```

END SELECT

```

```

' Lokale Variablen definieren.
' -----

```

```

BPos% = 0           ' Kopfposition am Band.
AZus$ = ""         ' Aktueller Zustand.
NZus$ = ""         ' Neuer Zustand.
AZei$ = ""         ' Aktuelles Zeichen.
NZeif$ = ""        ' Neues Zeichen.
Rich$ = ""         ' Neue Richtung.

```

```

PZeile% = 0        ' Zeile des Turing-Programmes.
Fehler% = 0        ' Fehlernummer für Fehler-Routine.

```

```

BPos% = VAL(BStart$) ' Der Lese- und Schreibkopf steht am Standard-
                    ' Startpunkt 99, oder, wenn der Benutzer einen
                    ' anderen Startpunkt angegeben hat, an diesem.

```

```

AZus$ = LEFT$(Prog(1), 5) ' Der in der Turing-Programmzeile 1 stehende
                          ' erste Zustand liefert den aktuellen Zustand.

```

```

' -----
' Eigentlicher Programmbereich des Turing-Interpreters.
' -----

```

```

ZeigeBand (BPos%)

```

```

DO ' Interpreterschleife.

```

```

PZeile% = 0          ' Es wird nach der Zeile gesucht, in der der
                    ' aktuelle Zustand und der tats„chliche, und das
DO ' Zeilensuchschleife. zu lesende Zeichen beide „bereinstimmen.

```

```

PZeile% = PZeile% + 1
IF PZeile% > PMax% THEN Fehler% = 1: EXIT DO
IF Prog(PZeile%) = "*****" THEN Fehler% = 1: EXIT DO

```

```

AZeif$ = MID$(Prog(PZeile%), 6, 1)

```

```

LOOP UNTIL (LEFT$(Prog(PZeile%), 5) = AZus$) AND (AZeif$ = Band(BPos%))

```

```

' Es kann sein, daß keine Turing-Programm-Zeile für den momentanen Zustand
' existiert, oder zumindest keine, die im momentanen Zustand das aktuell
' zu lesende Zeichen verarbeiten kann.

```

```

IF Fehler% = 1 THEN
ZeigeMeldung AZus$, BPos%, Rich$, 1
EXIT DO
END IF

```

```

' Es wurde eine Zeile gefunden, die den aktuellen Zustand und das aktuelle
' Zeichen enth„llt.

```

```

NZus$ = MID$(Prog(PZeile%), 7, 5) ' der neue Zustand wird bestimmt.
NZeif$ = MID$(Prog(PZeile%), 12, 1) ' das neue Zeichen wird bestimmt.
Rich$ = RIGHT$(Prog(PZeile%), 1) ' die neue Richtung wird bestimmt.

```

```

ZeigeBand (BPos%)
IF Trace$ <> "2" THEN
  LOCATE 8, 17, 0
  PRINT USING "###"; BPos%
END IF

```

```

IF Trace$ = "2" THEN
  ZeigeVariablen AZus$, NZus$, AZei$, NZei$, Rich$, BPos%
END IF

```

```

IF Trace$ > "0" THEN
  ZeigeProgramm PZeile%

```

```

  LiesZeichen In$
END IF

```

```

IF In$ = CHR$(27) THEN EXIT DO      ' falls Trace benutzt wird.
In$ = INKEY$                        ' falls Trace nicht benutzt wird, kann
IF In$ = CHR$(27) THEN EXIT DO      ' trotzdem abgebrochen werden.

```

```

' Band(BPos%) = "#"                  ' Das aktuelle Zeichen wird ausgelesen.
' Diese Anweisung w„re zwar der Funktion eines Turing-Programmes folgend,
' notwendig, wird aber durch die n„chste Anweisung unwichtig.
Band(BPos%) = NZei$                 ' Das neue Zeichen wird geschrieben.
AZus$ = NZus$                       ' Der neue Zustand wird zum aktuellen.

```

```

SELECT CASE Rich$

  CASE "l"
    BPos% = BPos% + 1
    IF BPos% = BMax% + 1 THEN BPos% = 1

  CASE "r"
    BPos% = BPos% - 1
    IF BPos% = 0 THEN BPos% = BMax%

  CASE "s"
    EXIT DO

  CASE ELSE
    ZeigeMeldung AZus$, BPos%, Rich$, 2
    EXIT DO

```

```

END SELECT

```

```

LOOP

```

```

ZeigeBand (BPos%)

```

```

LOCATE 25, 1, 0

```

```

Farbe 7, 1

```

```

PRINT " [Esc] f♦r R♦ckkehr zum Men♦ Das Turing-Programm ist abgearbeitet.
";

```

```

DO

```

```

  LiesZeichen In$

```

```

LOOP UNTIL In$ = CHR$(27)

```

```

LoescheBereich

```

END SUB

SUB Test (Teil%)

' In dieser Prozedur werden Testdaten in die Band- bzw. Programmdatei
' geschrieben. Dies wird ausgeführt, wenn der Benutzer keine Band- bzw.
' Programmdatei im Menü Voreinstellung vorgenommen hat.

' Informationen zum Format des Turing-Programms.

' a = Aktueller Zustand (5 Zeichen),
' b = Aktuelles Zeichen (1 Zeichen),
' c = Neuer Zustand (5 Zeichen),
' d = Neues Zeichen (1 Zeichen),
' e = Neue Richtung (1 Zeichen).

' Format: aaaaabccccde

IF Teil% = 1 THEN

' Programm.

' -----

' Beispiel: Addition zweier Zahlen.

Prog(1) = "s0 #s0 #1"

Prog(2) = "s0 1s1 1l"

Prog(3) = "s1 1s1 1l"

Prog(4) = "s1 0s2 1l"

Prog(5) = "s2 1s2 1l"

Prog(6) = "s2 #s3 #r"

Prog(7) = "s3 1s4 #r"

Prog(8) = "s4 1s5 #s"

Prog(9) = "*****" ' Programmendezeiger. Es ist für den eingebauten
' Editor sinnvoll, daß diese Zeile vorhanden ist,
' da sonst das Zeilenanfangen umständlich wird.

FOR I% = 10 TO PMax%

Prog(I%) = STRING\$(13, 0)

NEXT I%

ELSEIF Teil% = 2 THEN

' Band.

' -----

' Beispiel: Die Zahlen 3 und 3.

FOR I% = 1 TO BMax% ' Band mit leeren Feldern füllen.

Band(I%) = "#"

NEXT I%

Band(100) = "1" ' Jede Zahl n besteht aus n+1 Einsern.

Band(101) = "1"

Band(102) = "1"

Band(103) = "1"

Band(104) = "0"

Band(105) = "1"

Band(106) = "1"

Band(107) = "1"

Band(108) = "1"

END IF

END SUB

SUB Voreinstellung (Trace\$, TBName\$, TPName\$, BStart\$, Modus%)

```
' Zun chst werden die Trace-Stufe, die Band- und Programmnamen und
' der Startpunkt geladen.
' Dann k nnen - je nach Modus - diese Daten editiert werden.

' Zuerst werden die gespeicherten Werte geladen und in die entsprechenden
' Variablen eingeordnet.
```

```
KEY(1) OFF
```

```
OPEN "R", #1, "Turing2.Cfg", 1
FIELD #1, 1 AS Cfgt$
```

```
I% = 1
J% = 1
Trace$ = ""
TBName$ = ""
TPName$ = ""
BStart$ = ""
```

```
DO UNTIL EOF(1)
```

```
    GET #1, I%
    IF Cfgt$ = "#" THEN
        J% = J% + 1
        Cf$ = ""
    ELSE
        Cf$ = Cfgt$
    END IF
```

```
    SELECT CASE J%
```

```
        CASE 1
            IF LEN(Trace$) < 1 THEN Trace$ = Trace$ + Cf$
```

```
        CASE 2
            IF LEN(TBName$) < 61 THEN TBName$ = TBName$ + Cf$
```

```
        CASE 3
            IF LEN(TPName$) < 61 THEN TPName$ = TPName$ + Cf$
```

```
        CASE 4
            IF LEN(BStart$) < 4 AND INSTR("0123456789", Cf$) THEN BStart$ =
BStart$ + Cf$
```

```
        CASE ELSE
    END SELECT
```

```
    I% = I% + 1 ' N chstes Zeichen lesen.
```

```
LOOP
```

```
CLOSE #1
```

```
KEY(1) ON
```

```
' Hier wird die Prozedur verlassen, wenn nichts editiert werden soll.
```

```
IF Modus% = 0 THEN EXIT SUB
```

```
' Sonst wird das Fenster ge ffnet und die Daten werden angezeigt.
```

```
KEY(1) OFF
```

```
Farbe 12, 0
```

```

FOR I% = 2 TO 16
  Rahmen 11 - (INT(I% / 3)), 36 - I%, 10 + (INT(I% / 3)), 44 + I%, 0
  J% = I% - 1
  Rahmen 11 - (INT(J% / 3)), 36 - J%, 10 + (INT(J% / 3)), 44 + J%, 2
NEXT I%

```

```
KEY(1) ON
```

```

IF Trace$ < "0" THEN Trace$ = "0"
IF BStart$ < "1" THEN BStart$ = "99"

```

```
Farbe 14, 0
```

```

LOCATE 8, 23, 0
PRINT "Trace-Stufe      : "; Trace$; " "
LOCATE 10, 23, 0
PRINT "Turing-Band      : "; LEFT$(TbName$, 16); " "
LOCATE 11, 23, 0
PRINT "Turing-Programm : "; LEFT$(TpName$, 16); " "
LOCATE 13, 23, 0
PRINT "Band-Startpunkt  : "; BStart$; " "

```

```
Farbe 9, 0
```

```

LOCATE 8, 24, 0
PRINT "r"
LOCATE 10, 30, 0
PRINT "B"
LOCATE 11, 30, 0
PRINT "P"
LOCATE 13, 28, 0
PRINT "S"

```

```
Farbe 7, 1
```

```

LOCATE 25, 1, 0
PRINT " Tasten [R] [B] [P] [S] f r die Men punkte      [Esc] f r Abbruch
[F1] Hilfe ";

```

' Nun k nnen Sie editiert werden.

```
DO
```

```
LiesZeichen In$
```

```

SELECT CASE In$
  CASE "R", "r"
    Farbe 7, 1
    EditZeile 8, 41, 1, 1, Trace$, 0
    LOCATE 8, 41, 0
    Farbe 14, 0
    IF Trace$ > "2" THEN Trace$ = "2"
    IF Trace$ < "0" THEN Trace$ = "0"
    PRINT Trace$; " "
  CASE "B", "b"
    Farbe 7, 1
    EditZeile 10, 41, 16, 60, TbName$, 1
    LOCATE 10, 41, 0
    Farbe 14, 0
    PRINT LEFT$(TbName$, 16); " ";
    IF LEN(TbName$) < 17 THEN PRINT SPACE$(17 - LEN(TbName$))
  CASE "P", "p"
    Farbe 7, 1
    EditZeile 11, 41, 16, 60, TpName$, 1

```



```

        LOCATE 11, 41, 0
        Farbe 14, 0
        PRINT LEFT$(TPName$, 16); " ";
        IF LEN(TPName$) < 17 THEN PRINT SPACE$(17 - LEN(TPName$))
CASE "S", "s"
        Farbe 7, 1
        EditZeile 13, 41, 3, 3, BStart$, 0
        LOCATE 13, 41, 0
        Farbe 14, 0
        IF BStart$ < "1" THEN BStart$ = "99"
        IF VAL(BStart$) > BMax% THEN BStart$ = STR$(BMax%)
        PRINT BStart$;
        IF LEN(BStart$) < 4 THEN PRINT SPACE$(4 - LEN(BStart$))
CASE CHR$(27)
        EXIT DO
CASE ELSE
        END SELECT
LOOP

```

' Nun werden die aktuellen Werte gespeichert.

```
KEY(1) OFF
```

```
OPEN "R", #1, "Turing2.Cfg", 1
FIELD #1, 1 AS Cfgt$
```

```
Cfg$ = Trace$ + "#" + TBName$ + "#" + TPName$ + "#" + BStart$ + "#"
```

```
FOR I% = 1 TO LEN(Cfg$)
```

```
    RSET Cfgt$ = MID$(Cfg$, I%, 1)
```

```
    PUT #1, I%
```

```
NEXT I%
```

```
CLOSE #1
```

```
KEY(1) ON
```

```
LoescheBereich
```

```
END SUB
```

```
SUB ZeigeBand (BPos%)
```

' Diese Prozedur zeigt das Band in der Umgebung des Kopfes.

```
Farbe 14, 0
LOCATE 6, 3, 0
```

```
Start% = BPos% - 37
```

```
IF Start% < 1 THEN Start% = 1
```

```
FOR I% = Start% TO Start% + 74
```

```
    IF I% = BPos% THEN Farbe 14, 5 ELSE Farbe 14, 0
```

```
    IF I% < BMax% + 1 THEN
```

```
        PRINT Band(I%);
```

```
    ELSE
```

```
        PRINT Band(I% - BMax%);
```

```
    END IF
```

```
NEXT I%
```

```
END SUB
```

SUB ZeigeHilfe (Seite%)

' Diese Prozedur blendet Hilfsfunktionen ein.
' Sie verwendet nicht die globalen Variablen VCol% und HCol%,
' damit die Farben nach Verlassen dieser Prozedur wieder stimmen.

PCOPY 0, 1

COLOR 0, 2
Rahmen 6, 40, 23, 79, 2

LOCATE 25, 1, 0
PRINT " Taste [Esc] zum Verlassen der Hilfe
";

SELECT CASE HSeite%

CASE 1

H\$ = "Beim ersten Aufruf dieses Programmes&werden im aktuellen
Verzeichnis des&"

H\$ = H\$ + "aktuellen Laufwerkes bzw. der&aktuellen Platte zwei
Konfigurations-&"

H\$ = H\$ + "dateien erzeugt.&&Darin wird jeweils der Name und der&"

H\$ = H\$ + "Pfad folgender Dateien gespeichert:&"

H\$ = H\$ + "das Turing-Band und Turing-Programm,&"

H\$ = H\$ + "der Editor f♦r das Band und&der Editor f♦r das

Programm.&"

H\$ = H\$ + "Auch die Trace-Stufe und der Start-&punkt im Band werden
vermerkt.&"

CASE 2

H\$ = "Sofern bereits Eintr„ge in die&Konfigurationsdatei
Turing1.Cfg&"

H\$ = H\$ + "vorgenommen wurden, werden die-&se nun gelesen. Falls die
Angaben&"

H\$ = H\$ + "darin nicht zul„ssig sind, also&die Namen bzw. Pfade der
Band- bzw.&"

H\$ = H\$ + "Programmdateien nicht oder nicht&mehr existieren, wird
eine Fehler-&"

H\$ = H\$ + "meldung ausgegeben.&"

CASE 3

H\$ = "Sofern bereits Eintr„ge in die&Konfigurationsdatei
Turing2.Cfg&"

H\$ = H\$ + "vorgenommen wurden, werden die-&se nun gelesen. Falls die
Angaben&"

H\$ = H\$ + "darin nicht zul„ssig sind, also die&Namen bzw. Pfade der
Band- bzw.&"

H\$ = H\$ + "Programm-Editoren nicht oder nicht&mehr existieren, wird
eine Fehler-&"

H\$ = H\$ + "meldung ausgegeben.&"

CASE 4

H\$ = "In diesem Hauptmen♦ k„nnen die&folgenden Men♦punkte durch
Dr♦cken&"

H\$ = H\$ + "des jeweils farbig markierten&Buchstabens ausgew„hlt
werden:&&"

H\$ = H\$ + "Band editieren&Programm
editieren&Starten&Voreinstellung&"

H\$ = H\$ + "Optionen&Ende&&"

H\$ = H\$ + "Informationen zu den einzelnen&Men♦punkten k„nnen
abgerufen werden,&"

H\$ = H\$ + "nachdem sie ausgew„hlt wurden.&"

CASE 5

H\$ = "Band editieren:&Sie k"nnen das Turing-Band wahlweise&mit dem integrierten Editor&"
H\$ = H\$ + "oder einem externen Editor bearbeiten.&Dazu m"ssen Sie im Men" Optionen&"
H\$ = H\$ + "einen Namen mit dem dorthin f"hrenden&Pfad angeben.&"
H\$ = H\$ + "Zur Empfehlung: Benennen Sie die Band-&dateien immer als 'Name.TUR' .&"

CASE 6
H\$ = "Programm editieren:&Sie k"nnen das Turing-Programm wahl-&weise mit dem integrierten Editor&"
H\$ = H\$ + "oder einem externen Editor bearbeiten.&Dazu m"ssen Sie im Men" Optionen&"
H\$ = H\$ + "einen Namen mit dem dorthin f"hrenden&Pfad angeben.&"
H\$ = H\$ + "Zur Empfehlung: Benennen Sie die Pro-&grammdateien immer als 'Name.TUR' .&"

CASE 7
H\$ = "Starten:&Das Turing-Programm l"uft auf dem&Turing-Band ab, wobei je nach&"
H\$ = H\$ + "gew"hlter Trace-Stufe folgende&Informationen angezeigt werden:&&"
H\$ = H\$ + "0: Das Programm l"uft direkt ab.&"
H\$ = H\$ + "1: zus"tzlich Anzeige des Programmes.&"
H\$ = H\$ + "2: zus"tzlich Anzeige der Variablen.&"

CASE 8
H\$ = "Voreinstellung:&Die Auswahl der Men"punkte erfolgt&durch Dr"cken der markierten Tasten.&"
H\$ = H\$ + "Trace beeinfl"at den Ablauf des& Turing-Programmes,&"
H\$ = H\$ + "Turing-Band ist der vollst"ndige& Name des gew"nschten Bandes,&"
H\$ = H\$ + "Turing-Programm ist der vollst"ndige& Name des gew"nschten Programmes&"
H\$ = H\$ + "und Startpunkt gibt den Punkt an,& an dem der Kopf zu Anfang steht.&"

CASE 9
H\$ = "Optionen:&Die Auswahl der Men"punkte erfolgt&durch Dr"cken der markierten Tasten.&"
H\$ = H\$ + "Band-Editor ist der Name des Editors,& f"r das Turing-Band,&"
H\$ = H\$ + "Ba.-Verzeichnis ist der dorthin& f"hrende Pfad,&"
H\$ = H\$ + "Programm-Editor ist der Name des& Editors f"r das Turing-Programm,&"
H\$ = H\$ + "Pr.-Verzeichnis ist der dorthin& f"hrende Pfad.&"

CASE 10
H\$ = "Ende:&Dr"cken sie die Taste&"
H\$ = H\$ + "[J] zum Verlassen des Programmes.&"
H\$ = H\$ + "[N] um weiterzuarbeiten.&"

CASE 11
H\$ = "Laufzeitfehler 1:&Es existiert keine Programmzeile,&die das Element, an dem der Kopf&"
H\$ = H\$ + "gerade steht, im aktuellen Zustand&verarbeiten kann. "berpr"fen Sie&"
H\$ = H\$ + "das Programm mittels Trace, und&korrigieren Sie gegebenenfalls das&"
H\$ = H\$ + "Band bzw. das Programm.&"

CASE 12
H\$ = "Laufzeitfehler 2:&W"hrend bei der Benennung der Zu-&st"nde und

```

der Elemente keine&"
    H$ = H$ + "Einschränkungen gemacht werden, müssen die
Richtungsanweisungen&"
    H$ = H$ + "unbedingt i {l,r,s} sein, wobei"
    H$ = H$ + "l eine Bewegung nach links, &r eine Bewegung nach rechts
und&"
    H$ = H$ + "s den Programmstop hervorruft.&"

    CASE 13
    H$ = "Im Menü Voreinstellung kann man unter Turing-Band den
vollständigen Namen&"
    H$ = H$ + "des gewünschten Bandes eingeben, mit dem man arbeiten
will.&"
    H$ = H$ + "Die Pfadangabe muß vollständig sein.&"

    CASE 14
    H$ = "Im Menü Voreinstellung kann man unter Turing-Programm den
vollständigen&"
    H$ = H$ + "Namen des gewünschten Programmes eingeben, mit dem man
arbeiten will.&"
    H$ = H$ + "Die Pfadangabe muß vollständig sein.&"

    CASE 15
    H$ = "Im Menü Optionen kann man unter Band-Editor den Namen des
gewünschten&"
    H$ = H$ + "Editors mit evtl. notwendigen Parametern eingeben.&"
    H$ = H$ + "Unter Ba.-Verzeichnis muß dann der Pfad eingegeben
werden, der zu dem&"
    H$ = H$ + "Editor führt.&"
    H$ = H$ + "Die Pfadangabe muß vollständig sein.&"

    CASE 16
    H$ = "Im Menü Optionen kann man unter Programm-Editor den Namen des
ge-&"
    H$ = H$ + "wünschten Editors mit evtl. notwendigen Parametern
eingeben.&"
    H$ = H$ + "Unter Pr.-Verzeichnis muß dann der Pfad eingegeben
werden, der zu dem&"
    H$ = H$ + "Editor führt.&"
    H$ = H$ + "Die Pfadangabe muß vollständig sein.&"

    CASE ELSE
END SELECT

ZeigeText 7, 41, H$

DO
    LiesZeichen In$
LOOP UNTIL In$ = CHR$(27)

PCOPY 1, 0

END SUB

SUB ZeigeMeldung (AZus$, BPos%, Rich$, Art%)

' In dieser Prozedur werden Fehlermeldungen oder spezielle
' Informationen, die nicht immer erscheinen, ausgegeben.
' Zu einigen Fehlern existiert jeweils eine passende Hilfeseite.
' Die Case-Fälle oberhalb 7 entsprechen den Fehlernummern, die der
' Quick-Basic Compiler in ERR zurückgibt.

Ablage% = HSeite%
HSeite% = Art% + 10

```

```
SELECT CASE Art%
```

```
  CASE 1, 2
```

```
    Farbe 7, 4
```

```
    Rahmen 12, 32, 23, 59, 2
```

```
    M$ = "    Laufzeitfehler&&In dem Zustand    " + AZus$ + "&"
```

```
    M$ = M$ + "kann an Position " + STR$(BPos%) + "&"
```

```
    IF Art% = 1 THEN
```

```
      M$ = M$ + "das Bandedement    " + Band(BPos%) + "&"
```

```
      M$ = M$ + "nicht verarbeitet werden,&da keine Programmzeile&"
```

```
      M$ = M$ + "f&#252;r diesen Fall vorge-&sehen ist. &#220;berpr&#252;fen Sie&"
```

```
      M$ = M$ + "das Programm bzw. Band.&"
```

```
    ELSE
```

```
      M$ = M$ + "die Anweisung    " + Rich$ + "&"
```

```
      M$ = M$ + "nicht verarbeitet werden,&da sie kein zul,&ssiges&"
```

```
      M$ = M$ + "Kommando darstellt. Nur&Buchstaben i {l,r,s}&"
```

```
      M$ = M$ + "sind erlaubt.&"
```

```
    END IF
```

```
    ZeigeText 13, 33, M$
```

```
  CASE 3
```

```
    Farbe 12, 0
```

```
    Rahmen 6, 20, 16, 60, 2
```

```
    Farbe 14, 0
```

```
    M$ = "Versionsnummer : 02101991&&Lizenznummer    : " + LNr$ + "&&"
```

```
    M$ = M$ + "Aufruf am    : " + MID$(DATE$, 4, 2) + "." + LEFT$(
```

```
(DATE$, 2) + "." + RIGHT$(DATE$, 4) + "&&"
```

```
    M$ = M$ + "Alle Rechte bei : Mocom 1991&"
```

```
    ZeigeText 8, 22, M$
```

```
  CASE 4, 5, 6, 7
```

```
    Farbe 7, 4
```

```
    Rahmen 6, 10, 15, 70, 2
```

```
    M$ = "Datenfehler: Es ist nicht m&#246;glich, "
```

```
    SELECT CASE Art%
```

```
      CASE 4
```

```
        HSeite% = 2
```

```
        M$ = M$ + "das gew&#252;nschte&Turing-Band zu laden.&Es"
```

```
      CASE 5
```

```
        HSeite% = 2
```

```
        M$ = M$ + "das gew&#252;nschte&Turing-Programm zu laden.&Es"
```

```
      CASE 6
```

```
        HSeite% = 3
```

```
        M$ = M$ + "den gew&#252;nschten&Editor f&#252;r das Turing-Band
```

```
zu starten.&Er"
```

```
      CASE 7
```

```
        HSeite% = 3
```

```
        M$ = M$ + "den gew&#252;nschten&Editor f&#252;r das Turing-
```

```
Programm zu starten.&Er"
```

```
    END SELECT
```

```
    M$ = M$ + " existiert entweder selbst nicht, oder&befindet sich in
```

```
einem anderen Verzeichnis.&"
```

```
    ZeigeText 8, 12, M$
```

```
    LiesZeichen In$
```

```
  ' Nun folgen die Nummern echter Fehlermeldungen des Compilers.
```

```
  ' Sie beziehen sich auf Ein-/Ausgabe-Fehler der Dateien.
```

```
  CASE 24
```

M\$ = "1Laufzeitfehler: Innerhalb der vorher festgelegten&"
M\$ = M\$ + "erwarteten Zeit ist kein Signal von dem ange-&"
M\$ = M\$ + "sprochenen Ger,,t gekommen.&"

CASE 25

M\$ = "1Ger,,tefehler: Ein angesprochenes Ger,,t hat einen&"
M\$ = M\$ + "Hardware-Fehler gemeldet.&"

CASE 52

M\$ = "0Dateifehler: Es ist entweder der Dateiname oder die&"
M\$ = M\$ + "Dateinummer falsch.&"
M\$ = M\$ + "Ändern Sie bitte die Bezeichnung(-en) für Ihre&"
M\$ = M\$ + "Dateien in den Men's Voreinstellung und Optionen.&"

CASE 53

M\$ = "0Dateifehler: Die gewünschte Datei konnte nicht ge-&"
M\$ = M\$ + "funden werden. Es ist entweder der Dateiname falsch,&"
M\$ = M\$ + "oder die Pfadangabe ist nicht korrekt.&"
M\$ = M\$ + "Überprüfen Sie bitte die Dateinamen und Pfade&"
M\$ = M\$ + "daraufhin, ob sie korrekt bezeichnet sind.&"

CASE 57

M\$ = "1Ger,,te-Ein-/Ausgabefehler: Beim Datentransfer ist ein&"
M\$ = M\$ + "kritischer Fehler aufgetreten.&"

CASE 61

M\$ = "1Disketten-/Plattenfehler: Das gewünschte Speicher-&"
M\$ = M\$ + "medium verfügt nicht mehr über genügend freien&"
M\$ = M\$ + "Speicherplatz, um die Operation zuzulassen.&"

CASE 64

M\$ = "0Dateinamensfehler: Der gewählte Dateiname ist nicht&"
M\$ = M\$ + "zulässig. Verwenden Sie bitte nur die vom Betriebs-&"
M\$ = M\$ + "system zugelassen Zeichen für Datei-Bezeichnungen.&"
M\$ = M\$ + "Ändern Sie dazu die Bezeichnung(-en) für Ihre Dateien&"
M\$ = M\$ + "in den Men's Voreinstellung und Optionen.&"

CASE 68

M\$ = "0Ger,,tefehler: Das gewünschte Ger,,t ist entweder nicht&"
M\$ = M\$ + "verfügbar oder es existiert nicht.&"

CASE 70

M\$ = "0Zugriffsfehler: Auf die gewünschte Datei kann oder&"
M\$ = M\$ + "darf nicht (schreibend) zugegriffen werden.&"
M\$ = M\$ + "Ändern Sie bitte die Bezeichnung(-en) für Ihre&"
M\$ = M\$ + "Dateien in den Men's Voreinstellung und Optionen.&"

CASE 71

M\$ = "0Ger,,tefehler: Das gewünschte Ger,,t ist nicht bereit.&"
M\$ = M\$ + "Bitte überprüfen Sie, ob das Diskettenlaufwerk ver-&"
M\$ = M\$ + "riegelt ist, oder ob die Diskette korrekt eingelegt&"
M\$ = M\$ + "ist, oder evtl. im falschen Laufwerk liegt."

CASE 72

M\$ = "0Ger,,tefehler: Die Diskette/Festplatte ist beschädigt.&"
M\$ = M\$ + "Für Disketten: Überprüfen Sie, ob die Diskette evtl.&"
M\$ = M\$ + "noch nicht oder nicht korrekt formatiert ist, oder ob&"
M\$ = M\$ + "sie für diesen Laufwerkstyp falsch formatiert ist,&"
M\$ = M\$ + "oder ob sie nicht verwendbar ist.&"
M\$ = M\$ + "Für Festplatten: Lassen Sie Ihre Festplatte durch&"
M\$ = M\$ + "eine fachkundige Person überprüfen. Brechen Sie das&"
M\$ = M\$ + "Programm ab - Weiterarbeit ist nicht möglich."

CASE 75

```

M$ = "1Pfad-/Dateizugriffsfehler: Der Zugriff auf die Datei&"
M$ = M$ + "über diesen Pfad konnte vom Betriebssystem nicht&"
M$ = M$ + "durchgeführt werden.&"

CASE 76
M$ = "0Pfadfehler: Der Pfad konnte nicht gefunden werden.&"
M$ = M$ + "Ändern Sie bitte die entsprechenden Pfadangabe&"
M$ = M$ + "in den Menü's Voreinstellung und Optionen.&"

CASE ELSE
M$ = "1Unerwarteter Fehler: Der Grund für diesen Fehler ist&"
M$ = M$ + "nicht bekannt. Der Fehlercode ist: " + STR$(ERR) + ".&"
END SELECT

' Es folgt die spezielle Behandlung von Systemfehlern.

Sys$ = LEFT$(M$, 1)

IF Sys$ = "1" THEN
M$ = "Systemfehler !&&" + RIGHT$(M$, LEN(M$) - 1)
M$ = M$ + "&Das Programm muß abgebrochen werden.&"
Farbe 7, 4
Rahmen 7, 10, 20, 70, 2
ZeigeText 9, 12, M$
Art% = 1
END

ELSEIF Sys$ = "0" THEN
M$ = "Dateifehler!&&" + RIGHT$(M$, LEN(M$) - 1)
M$ = M$ + "&Die Konfigurationsdateien müssen neu gesetzt werden.&"
Farbe 7, 4
Rahmen 7, 10, 20, 70, 2
ZeigeText 9, 12, M$
Neu "Turing1.Cfg"      ' Nun werden die Konfigurationsdateien geleert,
Neu "Turing2.Cfg"
Art% = 0

END IF

END SUB

SUB ZeigeMenue (TeilNummer%)

' Diese Prozedur stellt die Bildschirmmaske dar.
' Je nach übergebener Nummer wird verschiedenes ausgeführt.

IF TeilNummer% = 0 THEN
SCREEN 0, 0, 0
Farbe 0, 3
CLS
Farbe 7, 1
Rahmen 1, 1, 3, 80, 2
END IF

IF TeilNummer% = 0 OR TeilNummer% = 1 THEN
Farbe 7, 1
LOCATE 2, 4, 0
PRINT "Band editieren  Programm editieren  Starten  Voreinstellung  Optionen
Ende"
END IF

IF TeilNummer% = 0 THEN
ZeigeTitel

```

```
END IF
```

```
IF TeilNummer% = 0 OR TeilNummer% = 2 THEN
```

```
  LOCATE 25, 1, 0
```

```
  Farbe 7, 1
```

```
  PRINT " Tasten [B] [P] [S] [V] [O] [E] f r die Men punkte
```

```
[F1] Hilfe ";
```

```
  Farbe 12, 1
```

```
  LOCATE 2, 4, 0
```

```
  PRINT "B"
```

```
  LOCATE 2, 20, 0
```

```
  PRINT "P"
```

```
  LOCATE 2, 40, 0
```

```
  PRINT "S"
```

```
  LOCATE 2, 49, 0
```

```
  PRINT "V"
```

```
  LOCATE 2, 65, 0
```

```
  PRINT "O"
```

```
  LOCATE 2, 75, 0
```

```
  PRINT "E"
```

```
END IF
```

```
END SUB
```

```
SUB ZeigeProgramm (PZeile%)
```

```
' Diese Prozedur stellt 10 Programmzeilen des Turing-Programmes dar,  
' wobei die aktuelle Programmzeile farbig hervorgehoben wird.
```

```
Farbe 11, 0
```

```
Start% = PZeile% - 5
```

```
IF Start% < 1 THEN Start% = 1
```

```
FOR I% = Start% TO Start% + 9
```

```
  IF Prog(I%) = "*****" THEN
```

```
    LOCATE 13 + I% - Start%, 62
```

```
    PRINT " "
```

```
    EXIT SUB
```

```
  END IF
```

```
  LOCATE 13 + I% - Start%, 62
```

```
  IF I% = PZeile% THEN Farbe 14, 5
```

```
  PRINT LEFT$(Prog(I%), 5); " "; MID$(Prog(I%), 6, 1); " ";
```

```
  PRINT MID$(Prog(I%), 7, 5); " "; MID$(Prog(I%), 12, 1); " ";
```

```
  PRINT RIGHT$(Prog(I%), 1)
```

```
  IF I% = PZeile% THEN Farbe 11, 0
```

```
NEXT I%
```

```
END SUB
```

```
SUB ZeigeText (XPos%, YPos%, Text$)
```

```
' In dieser Prozedur wird ein  bergabener Text zeilenweise ausgegeben,  
' wobei das kaufm nnische Und-Zeichen & als Zeilen-End-Zeichen verwendet  
' wird.
```

```
Zeile$ = ""
```

```
FOR I% = 1 TO LEN(Text$)
```

```
  T$ = MID$(Text$, I%, 1)
```

```
  IF T$ = "&" THEN
```

```
    LOCATE XPos%, YPos%, 0
```

```
    PRINT Zeile$
```



```

    Zeile$ = ""
    XPos% = XPos% + 1
ELSE
    Zeile$ = Zeile$ + T$
END IF
NEXT I%

END SUB

SUB ZeigeTitel

' Diese Prozedur zeigt den Programmtitel an.

Farbe 12, 0

KEY(1) OFF

FOR I% = 2 TO 20
    Rahmen 12 - (INT(I% / 3)), 40 - I%, 12 + (INT(I% / 3)), 40 + I%, 0
    J% = I% - 1
    Rahmen 12 - (INT(J% / 3)), 40 - J%, 12 + (INT(J% / 3)), 40 + J%, 2
NEXT I%

KEY(1) ON

Farbe 14, 0
LOCATE 8, 30, 0
PRINT "Turing - Interpreter"
LOCATE 10, 34, 0
PRINT "Version 2.10"
LOCATE 14, 25, 0
PRINT "Programmiert von Sebastian Mory"
LOCATE 16, 27, 0
PRINT "10.07.1991 Universit„t Ulm"

SLEEP

LoescheBereich

END SUB

SUB ZeigeVariablen (AZus$, NZus$, AZei$, NZei$, Rich$, BPos%)

Farbe 11, 0

LOCATE 13, 3, 0
PRINT USING "Aktueller Zustand: \  \"; AZus$

LOCATE 14, 3, 0
PRINT USING "Aktuelles Zeichen: \\"; AZei$

LOCATE 15, 3, 0
PRINT USING "Neuer Zustand      : \  \"; NZus$

LOCATE 16, 3, 0
PRINT USING "Neues Zeichen      : \\"; NZei$

LOCATE 17, 3, 0
PRINT USING "Neue Richtung      : \\"; Rich$

LOCATE 18, 3, 0
PRINT USING "Bandposition      : ###"; BPos%

END SUB

```

