

Anhang B

Tauchgangsimulation in GfA-Basic

```

DIM regler%(10,6)
DIM reglern$(10)

#####
# Deko-Simulation #
# M.Lardelli 1989 #
#####

DIM gewebe(16,2)      ! halbwertszeit, a, b,
FOR x%=1 TO 16
  READ gewebe(x%,0)
  READ gewebe(x%,1)
  READ gewebe(x%,2)
NEXT x%

DIM inertgdr gew(16)    ! inertgasdruck in den kompartimenten
DIM inertgdrtol(16,2)   ! tolerieter umgeb.druck, dekostufe, dekozeit

@init

      Hauptprogramm

REPEAT
  IF TIMER-timestate%>seclength% THEN    ! zeitberechnung
    IF tiefe%>0
      INC sec%
      timestate%=TIMER
    ELSE
      timestate%=TIMER
    ENDIF
  ENDIF
  IF sec%>=60 THEN
    INC min%
    sec%=sec%-60
    nmin!=TRUE
  ENDIF
  tiefe%=INT(regler%(0,5))
  hoehe%=INT(regler%(1,5))

  IF althoehe%<>hoehe%
    althoehe%=hoehe%
    @hoehenbereich
    GRAPHMODE 1
    TEXT 10,70,"BEREICH : "+STR$(ber%)+" "
    TEXT 10,50,"HHE : "+STR$(hoehe%)+" m..M "
  ENDIF
  IF alttiefe%<>tiefe%
    GRAPHMODE 1
    TEXT 10,130,"TIEFE : "+STR$(tiefe%)+" m "
    alttiefe%=tiefe%
  ENDIF

  umgebdruk=(tiefe%)/10+atmkomp          ! in bar
  inertgdrinsp=INT(100*(umgebdruk-0.063)*inertgasanteil)/100

```

```

IF nmin!           ! nur alle minute anzeigen
nmin!=FALSE

LINE altx,alty,320+min%*2,40+tiefe%*2    ! Grafik
altx=321+min%*2
alty=40+tiefe%*2

FOR x%=1 TO 16      ! Gewebespannungen, erlaubte Umgebungsdruecke
  k=0.69315/gewebe(x%,0)
  inertgdrgew(x%)=inertgdrgew(x%) +(inertgdrinsp-inertgdrgew(x%))*(1-EXP(-k))
  inertgdrtol(x%,0)=(inertgdrgew(x%)-gewebe(x%,1))*gewebe(x%,2)
  IF inertgdrtol(x%,0)=0
    inertgdrtol(x%,0)=0
  ENDIF
NEXT x%

GRAPHMODE 1
TEXT 10,110,"ZEIT      : "+STR$(min%)+" "
TEXT 10,90,"NULLZEIT : "+nullzeit$+" "
TEXT 10,150,"UMGEBUNGSDRUCK [bar] : "+STR$(umgebdruk)
TEXT 10,170,"INERTG.DRUCK (insp.) : "+STR$(inertgdrinsp)
TEXT 10,195,"INERTG.DRUCK (gewebe) : "
TEXT 10,270,"ERLAUBTER UMGEB.DRUCK : "
FOR x%=1 TO 16          ! n2-druecke,tolerierte dr. anzeige
igen
  TEXT 10+(x%-1)*40,220,STR$(x%)
  TEXT 5+(x%-1)*40,245,STR$(INT(inertgdrgew(x%)*100)/100)
  TEXT 5+(x%-1)*40,290,STR$(INT(inertgdrtol(x%,0)*100)/100)+" "
NEXT x%

TEXT 10,310,"DEKOSTUFE/DEKOZEIT/RESTNULLZEIT:" ! dekostufen berechnen

FOR x%=1 TO 16
  stufe=INT((inertgdrtol(x%,0)-atmdekomp+0.3)/0.3)*3
  inertgdrtol(x%,1)=stufe
  IF stufe<=0
    stufe$="-"
  ELSE
    stufe$=STR$(stufe)+" "
  ENDIF

  IF stufe>0          ! dekozeit berechnen
    k=0.69315/gewebe(x%,0)
    a=gewebe(x%,1)
    b=gewebe(x%,2)
    dekodruck=atmdekomp+0.1*(stufe-3)    ! umgebungsdruck nchsthhere dekostufe
    umgebdr=atmkomp+0.1*stufe
    gdrinsp=(umgebdr-0.063)*inertgasanteil
    pnull=inertgdrgew(x%)
    IF 1+(pnull-a-dekodruck/b)/(gdrinsp-pnull)>0
      inertgdrtol(x%,2)=LOG(1+(pnull-a-dekodruck/b)/(gdrinsp-pnull))/(-k)
    ENDIF
  ELSE
    inertgdrtol(x%,2)=0
  ENDIF
  IF inertgdrtol(x%,2)<>0
    zeit$=STR$(INT(inertgdrtol(x%,2)*10)/10)
  ENDIF
ENDFOR

```

```

ELSE
zeit$="-"
ENDIF
TEXT 5+(x%-1)*40,330,stufe$
TEXT 5+(x%-1)*40,350,zeit$
NEXT x%


tiefste=0
laengste=0
FOR x%=1 TO 16          ! tiefste dekostufe suchen
  IF inertgdrtol(x%,1)>tiefste
    tiefste=inertgdrtol(x%,1)
  ENDIF
NEXT x%
FOR x%=1 TO 16          ! lngste mit obiger tiefe
  IF inertgdrtol(x%,2)>laengste AND inertgdrtol(x%,1)=tiefste
    laengste=inertgdrtol(x%,2)
  ENDIF
NEXT x%


IF tiefste>0
  TEXT 10,390,"--> Unterste Dekostufe :"
  TEXT 250,390,STR$(tiefste)+" m "+STR$(INT(laengste+0.99))+"
ENDIF


restnullzeit berechnen
nullzeit=1000
x%=0
REPEAT
  INC x%
  k=0.69315/gewebe(x%,0)
  a=gewebe(x%,1)
  b=gewebe(x%,2)
  pnull=inertgdrgew(x%)
  dekodruck=atmdekomp      ! atmosphaerendruck
  pmax=dekodruck/b+a

  IF (inertgdrtol(x%,1)>0) ! schon deko -> keine Nullzeit
    nullzeitz=0
  ELSE
    IF inertgdrinsp>pmax AND pnull<pmax
      nullzeitz=LOG(1+(pnull-a-dekodruck/b)/(inertgdrinsp-pnull))/(-k)
    ELSE
      nullzeitz=1000      ! i<pmax,pnull<pmax --> unendlich
    ENDIF
  ENDIF

  IF nullzeitz<nullzeit ! krzeste NZ ist gltig
    nullzeit=nullzeitz
  ENDIF

  nullzeitz$=STR$(INT(nullzeitz))
  IF nullzeitz>=1000
    nullzeitz$=CHR$(223)
  ENDIF
  TEXT 5+(x%-1)*40,370,nullzeitz$+
UNTIL x%=16

```

```

    nullzeit=INT(nullzeit)      ! runden
    nullzeit$=STR$(nullzeit)
    IF nullzeit>=1000
        nullzeit$=CHR$(223)
    ENDIF
ENDIF

@asksiders
IF MOUSEK=2
    @init
ENDIF

UNTIL MOUSEK=3

END

'PROCEDURE hoehenbereich
IF hoehe%<3500
    atmkomp=0.74
    atmdekomp=0.67
    ber%=4
ENDIF
IF hoehe%<=2500
    atmkomp=0.84
    atmdekomp=0.76
    ber%=3
ENDIF
IF hoehe%<=1500
    atmkomp=0.93
    atmdekomp=0.86
    ber%=2
ENDIF
IF hoehe%<=700
    atmkomp=1.03
    atmdekomp=0.95
    ber%=1
ENDIF
RETURN

'PROCEDURE init
@hoehenbereich
CLS
PRINT
PRINT "          Dekompressions-simulation      M.Lardelli"
'
reglerpointer%=0
GOSUB initschieberegler(280,40,150,30,"DEPTH",0,65)
GOSUB initschieberegler(240,40,150,30,"HHE",0,4000)

timestate% = TIMER
'
sec%=0
min%=0
tiefe%=0
alttiefe%=0

```

```

althoehe%=-1
nullzeit$="0"
nullzeit=0
'
inertgasanteil=0.79
FOR x%=1 TO 16
    inertgdrgew(x%)=0.75
NEXT x%
'
FOR x%=1 TO 16
    inertgdrtol(x%,0)=0
    inertgdrtol(x%,1)=0
    inertgdrtol(x%,2)=0
NEXT x%
'
seclength%=3      ! normal: 200
nmin!=FALSE
'
LINE 320,190,600,190
LINE 320,190,320,40
DEFLINE 6
LINE 320,40,600,40
DEFLINE 1
LINE 600,190,600,40
altx=320
alty=40
RETURN
*****
PROCEDURE initschieberegler(x,y,lng,br,name$,def,skala)
    regler%(reglerpointer%,1)=x
    regler%(reglerpointer%,2)=y
    regler%(reglerpointer%,3)=lng
    regler%(reglerpointer%,4)=br
    regler%(reglerpointer%,5)=def
    regler%(reglerpointer%,6)=skala
    reglern$(reglerpointer%)=name$
    INC reglerpointer%
'
    BOX x,y,x+br,y+lng
    GOSUB printsilder(reglerpointer%-1)
RETURN
'
PROCEDURE printsilder(number)
    lng=regler%(number,3)
    lng2=lng-lng/10
    pos=regler%(number,5)*lng2/regler%(number,6)
    y1=pos-lng/20+regler%(number,2)+lng/20+1
    y2=pos+lng/20+regler%(number,2)+lng/20-1
    BOX regler%(number,1)+2,y1,regler%(number,1)-2+regler%(number,4),y2
RETURN
'
PROCEDURE asksliders
    MOUSE x,y,k
    IF k=1
        sel%=-1
        FOR i%=0 TO reglerpointer%
            IF x>regler%(i%,1) AND x<regler%(i%,1)+regler%(i%,4) AND y>regler%(i%,2)+r

```

```
egler%(i%,3)/20 AND y<regler%(i%,2)+regler%(i%,3)/20*19
    sel%=i%
ENDIF
NEXT i%
IF sel%>-1
    GRAPHMODE 3
    GOSUB printslder(sel%)
    qdef=(y-regler%(sel%,3)/20-regler%(sel%,2))*regler%(sel%,6)/(regler%(sel%,
3)/10*9)
        regler%(sel%,5)=qdef
        GOSUB printslder(sel%)
        event%=sel%
ELSE
    event%=-1
ENDIF
ELSE
    event%=-1
ENDIF
RETURN

' daten der 16 kompartimente: halbwertszeit, a,b

DATA 2.65, 2.2, 0.820
DATA 7.94, 1.5, 0.820
DATA 12.2, 1.05, 0.825
DATA 18.5, 0.9, 0.835
DATA 26.5, 0.75, 0.845
DATA 37, 0.6, 0.860
DATA 53, 0.45, 0.870
DATA 79, 0.43, 0.890
DATA 114, 0.43, 0.890
DATA 146, 0.43, 0.931
DATA 185, 0.43, 0.931
DATA 238, 0.35, 0.943
DATA 304, 0.23, 0.962
DATA 397, 0.23, 0.962
DATA 503, 0.23, 0.962
DATA 635, 0.23, 0.962
```

Anhang C

Programm zur Berechnung von Dekompressionstabellen nach Bühlmann

```

#####
# Deko-tabellenberechnung #
#####

maxtiefenstufe%=8
maxzeit%=60
maxdeko%=2      ! nur bis deko 6/4m tabelle berechnen

buehlmann=FALSE
hoehe%=0

IF buehlmann
  IF hoehe%>700
    dekostufe%=2      ! 12/9/6/3 oder 8/6/4/2
  ELSE
    dekostufe%=3
  ENDIF
  amerikanisch=TRUE      ! tiefenstufen in fuss(12/15/18/21) oder Meter(10/15/20)
  maxoxygen=FALSE      ! immer grsstmglicher o2-anteil im atemgas
  dekooxygen=FALSE      ! ab 6m dekompression mit reinem o2
  hoehenstufen=TRUE      ! verwenden von Hoehenstufe oder exakte berechnung
  sicherheitszuschlag%=2
ELSE
  dekostufe%=3      ! 12/9/6/3 oder 9/6/4/2
  amerikanisch=TRUE      ! tiefenstufen in fuss(12/15/18/21) oder Meter(10/15/20)
)
maxoxygen=FALSE      ! immer grsstmglicher o2-anteil im atemgas
dekooxygen=TRUE      ! ab 6m dekompression mit reinem o2
hoehenstufen=TRUE      ! verwenden von Hoehenstufe oder exakte berechnung
sicherheitszuschlag%=0
ENDIF
'

DIM gewebe(16,2)      ! halbwertszeit, a, b,
FOR x%=1 TO 16
  READ gewebe(x%,0)
  READ gewebe(x%,1)
  READ gewebe(x%,2)
NEXT x%

DIM inertgdrgew(16)      ! inertgasdruck in den kompartimenten
DIM inertgdrtol(16,2)      ! tolerieter umgeb.druck, dekostufe, dekozeit
DIM stufen%(8)      ! 1-8 = stufen 3m - 24m

DIM tiefe(20)
IF amerikanisch
  tiefe(1)=9
  tiefe(2)=12

```

```

tiefen(3)=15
tiefen(4)=18
tiefen(5)=21
tiefen(6)=24
tiefen(7)=27
tiefen(8)=30
tiefen(9)=33
tiefen(10)=36
tiefen(11)=39
tiefen(12)=42
tiefen(13)=45
tiefen(14)=48
tiefen(15)=51
ELSE
    tiefen(1)=10
    tiefen(2)=15
    tiefen(3)=20
    tiefen(4)=25
    tiefen(5)=30
    tiefen(6)=35
    tiefen(7)=40
    tiefen(8)=45
    tiefen(9)=50
    tiefen(10)=55
    tiefen(11)=60
    tiefen(12)=65
    tiefen(13)=70
    tiefen(14)=75
    tiefen(15)=80
ENDIF
;

@init
;

PRINT
PRINT "Hoehe : ";STR$(hoehe%);" m..M      Bereich : ";STR$(ber%)
PRINT
INPUT "Drucker/Bildschirm (D/B)      :",dev$
lev$=UPPER$(dev$)
;
IF dev$="D"
    LPRINT
    LPRINT "Hoehe : ";STR$(hoehe%);" m..M      Bereich : ";STR$(ber%)
    IF maxoxygen
        LPRINT "Tauchen mit maximalem O2-Anteil"
    ENDIF
    IF dekoxygen
        LPRINT "Dekompression ab 6m mit reinem O2"
    ENDIF
    LPRINT "Sicherheitszuschlag zur Tiefe : ";sicherheitszuschlag%;"m"
    LPRINT
    LPRINT
    LPRINT " Tiefe   Zeit      ";
    FOR x%=8 DOWNTO 1
        LPRINT USING "###      ",x%*dekostufe%;
    NEXT x%
    LPRINT
    LPRINT

```

```

ENDIF

'OR o%=1 TO maxtiefenstufe%
t%=tiefe(o%)
CLS
PRINT " Tiefe: ";t%
PRINT " Tiefe Zeit      ";
FOR x%=8 DOWNTO 1
    PRINT USING "###      ",x%*dekostufe%;
NEXT x%
PRINT
PRINT
'

t%=t%+sicherheitszuschlag%
@nullzeitberechnen(t%)
PRINT " Nullzeit : ";nullzeit
PRINT
IF dev$="D"
    LPRINT
    IF nullzeit<>999
        nzt$=STR$(nullzeit)+"   "
    ELSE
        nzt$=" ---"
    ENDIF
    LPRINT " ";t%;" m    Nullzeit = ";nzt$
ENDIF
'

FOR z%=5 TO maxzeit% STEP 5
    @tauchgang(t%,z%)
NEXT z%
'

IF dev$<>"D"
    REPEAT
        UNTIL INKEY$<>"""
ENDIF
NEXT o%
END

'

PROCEDURE nullzeitberechnen(tiefe%)
LOCAL x%,a,b,stufe,k,dekodruck,umgebdruk,inertgdrinsp,pnull
x%=0
@tauchganginit
nullzeit=1000
REPEAT
    INC x%
    k=0.69315/gewebe(x%,0)
    a=gewebe(x%,1)
    b=gewebe(x%,2)
    pnull=inertgdrgew(x%)
    dekodruck=atmdekomp
    '
    umgebdruk=(tiefe%)/10+atmkomp          ! in bar
    IF maxoxygen
        inertgasanteil=1-1.6/umgebdruk
    ELSE
        inertgasanteil=0.79
    ENDIF

```

```

IF inertgasanteil<0
  inertgasanteil=0
ENDIF
inertgdrinsp=(umgebdruck-0.063)*inertgasanteil
'
pmax=dekodruck/b+a

IF (inertgdrtol(x%,1)>0)
  nullzeitx=0
ELSE
  IF inertgdrinsp>pmax AND pnull<pmax
    nullzeitx=LOG(1+(pnull-a-dekodruck/b)/(inertgdrinsp-pnull))/(-k)
  ELSE
    nullzeitx=1000
  ENDIF
ENDIF

IF nullzeitx<nullzeit
  nullzeit=nullzeitx
ENDIF
UNTIL x%=16

nullzeit=INT(nullzeit-0.5)
RETURN

PROCEDURE tauchgang(maxtief%,zeit%)
LOCAL m%,z%
m%=maxtief%
z%=zeit%
@tauchganginit
drucken=FALSE
REPEAT
  @tauchen(maxtief%,zeit%)
  ' @Auftauchen(Maxtief%,Tiefste) ! zum nchsten dekostop
  IF NOT nodeko
    stufen%(tiefste/dekostufe%)=INT(laengste+1)
    ' Print Tiefste,Stufen%(Tiefste/3-1)
    drucken=TRUE
  ENDIF
  maxtief%=tiefste
  zeit%=INT(laengste+1)
UNTIL nodeko
IF drucken
  PRINT USING "### m   ",m%-sicherheitszuschlag%;
  PRINT USING "### '     ",z%;
  IF dev$="D"
    LPRINT USING "### m   ",m%;
    LPRINT USING "### '     ",z%;
  ENDIF
  @stufenprinten           ! ausdrucken der dekostufen/zeiten
  PRINT
  IF dev$="D"
    LPRINT
  ENDIF
ENDIF
RETURN

```

```

PROCEDURE stufenprinten
LOCAL x%
FOR x%=8 DOWNTO 1
  IF stufen%(x%)>0
    PRINT USING "###      ",stufen%(x%);
    IF dev$="D"
      LPRINT USING "###      ",stufen%(x%);
    ENDIF
  ELSE
    PRINT " -      ";
    IF dev$="D"
      LPRINT " -      ";
    ENDIF
  ENDIF
NEXT x%
RETURN

PROCEDURE auftauchen(von,bis) ! entstiegung whrend des auftauchens bercks.
LOCAL diff,tiefe
diff=von-bis
auftauchzeit=diff/10 ! 10m/min
FOR tiefe=von+diff/10 TO bis-diff/10 STEP diff/10 ! in 10 schritten berechnen
  @tauchen(tiefe,auftauchzeit/8)
NEXT tiefe
RETURN

PROCEDURE tauchen(tiefe%,zei%)
LOCAL min%,umgeldruck
umgeldruck=(tiefe%)/10+atmkomp ! in bar
IF maxoxygen
  inertgasanteil=1-1.6/umgeldruck
ELSE
  inertgasanteil=0.79
ENDIF
IF tiefe%<6 AND dekooxygen
  inertgasanteil=0
ENDIF
IF inertgasanteil<0
  inertgasanteil=0
ENDIF
inertgdrinsp=(umgeldruck-0.063)*inertgasanteil
@geweberechnen(zei%)
@dekostufen
@tiefste
RETURN

PROCEDURE geweberechnen(zei%)
LOCAL k,x%
FOR x%=1 TO 16
  k=0.69315/gewebe(x%,0)
  inertgdr gew(x%)=inertgdr gew(x%)+(inertgdrinsp-inertgdr gew(x%))*(1-EXP(-k*zei%))
  inertgdrtol(x%,0)=(inertgdr gew(x%)-gewebe(x%,1))*gewebe(x%,2)
NEXT x%
RETURN

```

```

PROCEDURE dekostufen
  LOCAL x%,a,b,stufe,k,dekodruck,umgebdr,gdrinsp,pnull
  FOR x%=1 TO 16
    stufe=INT((inertgdrtol(x%,0)-atmdekomp+dekostufe%/10)/dekostufe%*10)*dekostu
    ie%
    inertgdrtol(x%,1)=stufe

    IF stufe>0          ! dekozeit berechnen
      k=0.69315/gewebe(x%,0)
      a=gewebe(x%,1)
      b=gewebe(x%,2)
      dekodruck=atmdekomp+0.1*(stufe-dekostufe%)                         ! umgebungsdruck nchst
      hhere dekostufe
      umgebdr=atmkomp+0.1*stufe
      IF maxoxygen
        inertganteil=1-1.6/umgebdr
      ELSE
        inertganteil=0.79
      ENDIF
      IF stufe<9 AND dekooxygen
        inertganteil=0.1      ! dekompression mit reinem sauerstoff ab 6m
      ENDIF
      IF inertganteil<0
        inertganteil=0
      ENDIF
      gdrinsp=(umgebdr-0.063)*inertganteil
      pnull=inertgdrgew(x%)
      IF 1+(pnull-a-dekodruck/b)/(gdrinsp-pnull)>0
        inertgdrtol(x%,2)=LOG(1+(pnull-a-dekodruck/b)/(gdrinsp-pnull))/(-k)
      ENDIF
    ELSE
      inertgdrtol(x%,2)=0
    ENDIF
  NEXT x%
  RETURN

```

```

PROCEDURE tiefste
  LOCAL x%
  tiefste=0
  laengste=-1
  FOR x%=1 TO 16          !tiefste
    IF inertgdrtol(x%,1)>tiefste
      tiefste=inertgdrtol(x%,1)
    ENDIF
  NEXT x%
  FOR x%=1 TO 16          ! lngste mit obiger tiefe
    IF inertgdrtol(x%,2)>laengste AND inertgdrtol(x%,1)=tiefste
      laengste=inertgdrtol(x%,2)
    ENDIF
  NEXT x%
  IF tiefste=0
    nodeko=TRUE
  ELSE
    nodeko=FALSE

```

```
ENDIF  
RETURN
```

```
'ROCEDURE hoehenbereich  
IF hoehe%< 3500  
    atmkomp=0.74  
    atmdekomp=0.67  
    ber%=4  
ENDIF  
IF hoehe%<=2500  
    atmkomp=0.84  
    atmdekomp=0.76  
    ber%=3  
ENDIF  
IF hoehe%<=1500  
    atmkomp=0.93  
    atmdekomp=0.86  
    ber%=2  
ENDIF  
IF hoehe%<=700  
    atmkomp=1.03  
    atmdekomp=0.95  
    ber%=1  
ENDIF  
IF NOT hoehenstufe  
    atmkomp=0.86  
    atmdekomp=0.8  
ENDIF  
RETURN
```

```
PROCEDURE tauchganginit  
LOCAL x%  
  
FOR x%=1 TO 8  
    stufen%(x%)=0  
NEXT x%  
  
FOR x%=1 TO 16  
    inertgdrgew(x%)=0.75  
NEXT x%  
  
FOR x%=1 TO 16  
    inertgdrtol(x%,0)=0  
    inertgdrtol(x%,1)=0  
    inertgdrtol(x%,2)=0  
NEXT x%  
RETURN
```

```
PROCEDURE init  
@hoehenbereich  
CLS  
PRINT  
PRINT "          Deko-tabellen Berechnung      M.Lardelli"
```

inertgasanteil=0.79

RETURN

' daten der 16 kompartimente: halbwertszeit, a,b

)ATA 2.65, 2.2, 0.820  
DATA 7.94, 1.5, 0.820  
DATA 12.2, 1.05, 0.825  
DATA 18.5, 0.9, 0.835  
DATA 26.5, 0.75, 0.845  
DATA 37, 0.6, 0.860  
DATA 53, 0.45, 0.870  
DATA 79, 0.43, 0.890  
DATA 114, 0.43, 0.890  
DATA 146, 0.43, 0.931  
DATA 185, 0.43, 0.931  
DATA 238, 0.35, 0.943  
DATA 304, 0.23, 0.962  
DATA 397, 0.23, 0.962  
DATA 503, 0.23, 0.962  
DATA 635, 0.23, 0.962

Anhang D

Beispiel für ein Dekocomputerprogramm in GfA-Basic

Beispielprogramm  
Deko-Computer  
(simulation)

M.lardelli 89

```
key$="""
?PRINT "q=quit"

DIM faktor(16,1)
DIM halbwertszeit(16)

FOR x%=1 TO 16
  READ halbwertszeit(x%)
  READ faktor(x%,0)
  READ faktor(x%,1)
NEXT x%

DIM inertgdr gew(16)

DIM inertgdrtol(16)
DIM dekostufe(16)
DIM dekozeit(16)

@init

REPEAT
  @tieflesen
  @zeit          ! interrupt

  IF nmin%=1
    @gewebeberechnen
    @suchlaengste
    @printdeko
  ENDIF

  @zeitprinten

  IF tief e% > alttiefe%
    @tiefeprinten
    alttiefe% = tiefe%
  ENDIF
  key$=INKEY$
UNTIL key$="q"
END

PROCEDURE printdeko
laengste=INT(laengste+0.99)
nullzeit=INT(nullzeit)
```

```

TEXT 10,110,"Dekostufe :" +STR$(tiefste) + " m "
TEXT 10,130,"Dekozeit :" +STR$(laengste) + " "
TEXT 10,160,"Nullzeit :" +STR$(nullzeit) + " "
RETURN

PROCEDURE gewebeberechnen
umgeldruck=tiefe%/10+atmkomp
inertgdrinsp=(umgeldruck-0.063)*inertgasanteil
|
nullzeit=1000
tiefste=0
|
FOR x%=1 TO 16
|
    allgemeine konstanten
|
k=0.69315/halbwertszeit(x%)
a=faktor(x%,0)
b=faktor(x%,1)
|
    gewebespannungen/tolerierter umgebungsdruck
|
inertgdr gew(x%)=inertgdr gew(x%)+(inertgdrinsp-inertgdr gew(x%))*(1-EXP(-k))
pnull=inertgdr gew(x%)
inertgdr tol(x%)=(pnull-a)*b
|
    dekostufen/zeiten
|
dekostufe(x%)=INT((inertgdr tol(x%)-atmdekomp+0.3)/0.3)*3
IF dekostufe(x%)>tiefste
    tiefste=dekostufe(x%)
ENDIF
|
IF dekostufe(x%)>0           !dekozeit berechnen
    dekodruck=atmdekomp+0.1*(dekostufe(x%)-3)
    dekoumgebdr=atmkomp+0.1*dekostufe(x%)
    dekogdrinsp=(dekoumgebdr-0.063)*inertgasanteil
    dekozeit(x%)=LOG(1+(pnull-a-dekodruck/b)/(dekogdrinsp-pnull))/(-k)
ELSE
    dekozeit(x%)=0
ENDIF
|
    nullzeit
|
dekodruck=atmdekomp
pmax=dekodruck/b+a
IF dekostufe(x%)>0
    nullzeit2=0
ELSE
    IF inertgdrinsp>pmax AND pnull<pmax
        nullzeit2=LOG(1+(pnull-a-dekodruck/b)/(inertgdrinsp-pnull))/(-k)
    ELSE
        nullzeit2=1000
    ENDIF
ENDIF
IF nullzeit2<nullzeit

```

```
    nullzeit=nullzeit2  
ENDIF
```

```
NEXT x%  
RETURN
```

```
PROCEDURE suchlaengste  
laengste=0  
  
FOR x%=1 TO 16  
    IF dekostufe(x%)=tiefste AND dekozeit(x%)>laengste  
        laengste=dekozeit(x%)  
    ENDIF  
NEXT x%  
RETURN
```

```
PROCEDURE zeitprinten  
IF nmin%=1  
    TEXT 10,90,"Zeit : "+STR$(min%)+" : "  
    nmin%=0  
ENDIF  
RETURN
```

```
PROCEDURE zeit  
IF TIMER-timestate%>seclength%  
    INC sec%  
    timestate%=TIMER  
ENDIF  
IF sec%>=60  
    INC min%  
    sec%=sec%-60  
    nmin%=1      ! neue minute-> anzeigen  
ENDIF  
RETURN
```

```
PROCEDURE init  
alttiefe%=-1  
tiefe%=0  
  
timestate%=TIMER  
sec%=0  
min%=0  
seclength%=4    ! timer initialisieren  
  
nullzeit=0  
  
inertgasanteil=0.79  
FOR x%=1 TO 16
```

```
inertgdr gew(x%)=0.75
inertgdr tol(x%)=0
dekostufe(x%)=0
dekozeit(x%)=0
NEXT x%
atmcomp=1.03           ! entsprechend 0-700 m..M
atmdekomp=0.95
RETURN
```

```
PROCEDURE tiefeprinten
  TEXT 10,70,"Tiefe: "+STR$(tiefe%)+" m"
RETURN
```

```
PROCEDURE tiefelesen
  IF key$=CHR$(0)+"H"      ! hoher
    tiefe%=tiefe%-1
    IF tiefe%<0
      tiefe%=0
    ENDIF
  ENDIF
  IF key$=CHR$(0)+"P"      ! tiefer
    tiefe%=tiefe%+1
    IF tiefe%>99
      tiefe%=99
    ENDIF
  ENDIF
RETURN
```

```
daten der 16 kompartimente: halbwertszeit, a,b
```

```
DATA 2.65, 2.2, 0.820
DATA 7.94, 1.5, 0.820
DATA 12.2, 1.05, 0.825
DATA 18.5, 0.9, 0.835
DATA 26.5, 0.75, 0.845
DATA 37, 0.6, 0.860
DATA 53, 0.45, 0.870
DATA 79, 0.43, 0.890
DATA 114, 0.43, 0.890
DATA 146, 0.43, 0.931
DATA 185, 0.43, 0.931
DATA 238, 0.35, 0.943
DATA 304, 0.23, 0.962
DATA 397, 0.23, 0.962
DATA 503, 0.23, 0.962
DATA 635, 0.23, 0.962
```

```
----- ENDE -----
```

Anhang E

Shell für Entwicklungssoftware in Turbo-Pascal

```
uses crt,dos;  
  
var  
  recfile:text;  
  s19file:text;  
  zeile:STRING[255];  
  
begin  
  Assign(recfile,ParamStr(1)+'.rec');  
  Assign(s19file,ParamStr(1)+'.s19');  
  Reset(recfile);  
  Rewrite(s19file);  
  
  ReadLn(recfile,zeile); {erste zeile weglassen}  
  WHILE NOT Eof(recfile) DO  
    BEGIN  
      ReadLn(recfile,zeile);  
      WriteLn(s19file,zeile);  
    END;  
  close(s19file);  
  Close(recfile);  
end.
```

```
PROGRAM assemblershell;

USES dos,crt,windows,popup,pulld;

TYPE string20=STRING[20];
      string40=STRING[40];

VAR choice:INTEGER;
    editor,assembler,objekt,
    source,assemblername,editorname,
    linkername,epromername,
    workfile:string40;
    config:TEXT;
    dirname,dirmaske:STRING[40];

PROCEDURE makepdmenus;
BEGIN
  initmenus(TRUE,TRUE);
  WITH pdmenu DO
  BEGIN
    menupunktanzahl:=7;
    titel[1]:='F-ile ';
    menuenr[1]:=1;
    titel[2]:='E-dit ';
    menuenr[2]:=2;
    titel[3]:='A-ss ';
    menuenr[3]:=3;
    titel[4]:='P-ROMer';
    menuenr[4]:=4;
    titel[5]:='S-imulator';
    menuenr[5]:=9;
    titel[6]:='O-ptions';
    menuenr[6]:=5;
    titel[7]:='H-help';
    menuenr[7]:=10;
  END;
  WITH menues[1] DO {file}
  BEGIN
    laenge:=5;
    menueeintr[1]:='S-ytemdatum';
    menueeintr[2]:='C-hange-Dir';
    menueeintr[3]:='D-irectory';
    menueeintr[4]:='O-S-Shell';
    menueeintr[5]:='Q-uit';
  END;
  WITH menues[2] DO {edit}
  BEGIN
    laenge:=5;
    menueeintr[1]:='E-xterner';
    menueeintr[2]:='I-nterner';
    menueeintr[3]:='-----';
    menueeintr[4]:='D-atei';
    menueeintr[5]:='';
  END;
  WITH menues[3] DO {ass}
  BEGIN
```

```
BEGIN
laenge:=10;
menueeintr[1]:="S-tart-Assembler";
menueeintr[2]:="A-ssembler";
menueeintr[3]:="-----";
menueeintr[4]:="I-Start-Linker";
menueeintr[5]:="L-inker";
menueeintr[6]:="-----";
menueeintr[7]:="M-ach alles selv";
menueeintr[8]:="-----";
menueeintr[9]:="Z-eigen Listing";
menueeintr[10]:="D-rucken Listing";
END;
WITH menues[4] DO          {eprom}
BEGIN
laenge:=5;
menueeintr[1]:="S-tart-Eepromer";
menueeintr[2]:="E-PROMer";
menueeintr[3]:="-----";
menueeintr[4]:="B-inrdatei zeigen";
menueeintr[5]:="";
END;
WITH menues[5] DO          {options}
BEGIN
laenge:=5;
menueeintr[1]:="      ";
menueeintr[2]:="E-nvironment";
menueeintr[3]:="D-irectories";
menueeintr[4]:="S-ave Options";
menueeintr[5]:="      ";
END;

WITH menues[6] DO          {options/directories}
BEGIN
laenge:=4;
menuettitel:="Direct";
menueeintr[1]:="E-ditor";
menueeintr[2]:="A-ss/Link";
menueeintr[3]:="O-bj.-Code";
menueeintr[4]:="S-source-Code";
END;
WITH menues[7] DO
BEGIN
laenge:=4;
menuettitel:="Envir";
menueeintr[1]:="T-on ein";
menueeintr[2]:="T-on aus";
menueeintr[3]:="      ";
menueeintr[4]:="      ";
END;
WITH menues[8] DO
BEGIN
laenge:=6;
menuettitel:="Maske";
menueeintr[1]:="1 *.asc";
menueeintr[2]:="2 *.lst";
menueeintr[3]:="3 *.rec";
menueeintr[4]:="4 *.bin";
```

```
menueeintr[5]:='5  *.*    ';
menueeintr[6]:='6  andere';
END;
WITH menues[9] DO
BEGIN
laenge:=10;
menueeintr[1]:='S-tart-Simulator  ';
menueeintr[2]:='E-dit Comm.-File  ';
menueeintr[3]:='-----  ';
menueeintr[4]:='D-ump  ';
menueeintr[5]:='G-eninp  ';
menueeintr[6]:='-----  ';
menueeintr[7]:='Convert .rec-.s19  ';
menueeintr[8]:='Z-eige .s19-File  ';
menueeintr[9]:='-----  ';
menueeintr[10]:='F-unktionstasten  ';
END;

WITH menues[10] DO
BEGIN
laenge:=5;
menueeintr[1]:='A-ssembler  ';
menueeintr[2]:='P-epromer  ';
menueeintr[3]:='E-ditor  ';
menueeintr[4]:='S-imulator  ';
menueeintr[5]:='F-unkt.tasten  ';
END;
initpulldown;
displaypulldown;
END;
```

```
FUNCTION getpfad(menutitel:string20):string40;
VAR pfade:string40;
BEGIN
OpenWindow(menutitel,5,5,45,8);
WriteLn;
Write('Pfad >');
ReadLn(pfad);
getpfad:=pfad;
END;
```

```
PROCEDURE saveoptions;
BEGIN
Rewrite(config);
WriteLn(config,workfile);
WriteLn(config,assemblername);
WriteLn(config,editorname);
WriteLn(config,linkername);
WriteLn(config,epromername);
WriteLn(config,dirname);
WriteLn(config,dirmaske);
IF soundeff THEN
WriteLn(config,'SOUND')
ELSE
```

```
    WriteLn(config,'NOSOUND');
    WriteLn(config,'KONFIGURATION ASSEMBLERSHELL M.Lardelli 89');
    Close(config);
END;

PROCEDURE dirchange;
BEGIN
  Exec ('\command.com', '/C cd '+dirname);
END;

PROCEDURE execute;
VAR taste:CHAR;
    wahl:INTEGER;

BEGIN
  CASE choice OF
    13:BEGIN          {directory}
      wahl:=displaymenu(8,40,6);
      CASE wahl OF
        1:dirmaske:='*.asc';
        2:dirmaske:='*.lst';
        3:dirmaske:='*.rec';
        4:dirmaske:='*.bin';
        5:dirmaske:='*.*';
        6:BEGIN
          OpenWindow('Directory',5,5,45,8);
          WriteLn('-----'+dirmaske+'-----');
          Write('Maske >');
          ReadLn(dirmaske);
          CloseWindow;
        END;
      END;
      removelastmenu;
      ClrScr;
      Exec ('\command.com', '/C ddir '+dirmaske);
      taste:=ReadKey;
      displaypulldown;
    END;
    14:BEGIN          {os-shell}
      ClrScr;
      WriteLn;
      WriteLn('EXIT = Rcksprung zur Shell');
      Exec ('\command.com', '');
      displaypulldown;
    END;
    11:printdate;

    21:BEGIN          {editor}
      Exec ('\command.com', '/C Key-Fake "ne'+workfile+'.asc" 13');
      Exec ('\command.com', '/C '+editorname);
      displaypulldown;
    END;

24:BEGIN
```

```
OpenWindow('Arbeitsdatei',5,5,45,8);
WriteLn('-----'+workfile+'-----');
Write('Name >');
ReadLn(workfile);
CloseWindow;
display pulldown;
END;

31:BEGIN          {assembler}
ClrScr;
Exec ('\command.com', '/C '+assemblername+' '+workfile+'.asc');
WriteLn;
WriteLn;
Write('Fertig. Irgendeine Taste drcken >');
taste:=ReadKey;
display pulldown;
END;

32:BEGIN
OpenWindow('Assembler',5,5,45,8);
WriteLn('-----'+assemblername+'-----');
Write('Name >');
ReadLn(assemblername);
CloseWindow;
display pulldown;
END;

34:BEGIN
ClrScr;
Exec ('\command.com', '/C '+linkername+' '+workfile+'.rec');
WriteLn;
WriteLn;
Write('Fertig. Irgendeine Taste drcken >');
taste:=ReadKey;
display pulldown;
END;

35:BEGIN
OpenWindow('Linker',5,5,45,8);
WriteLn('-----'+linkername+'-----');
Write('Name >');
ReadLn(linkername);
CloseWindow;
display pulldown;
END;

37:BEGIN
ClrScr;
WriteLn('Assemblieren , .BIN-Datei erstellen.');
WriteLn;
Exec ('\command.com', '/C '+assemblername+' '+workfile+'.asc');
WriteLn;
WriteLn;
Exec ('\command.com', '/C '+linkername+' '+workfile+'.rec');
WriteLn;
Write('Fertig. Irgendeine Taste drcken >');
taste:=ReadKey;
display pulldown;
END;
```

```
39:BEGIN
    Exec('\'command.com', '/C browse '+workfile+'.lst');
END;

40:BEGIN
    ClrScr;
    WriteLn;WriteLn;
    WriteLn("Listing drucken .... ");
    Exec('\'command.com', '/C type '+workfile+'.lst>lpt1:');
    displaypulldown;
END;

41:BEGIN
    Exec('\'command.com', '/C Key-Fake "2'+workfile+'.bin" 13 "O" 13 13');
    Exec('\'command.com', '/C '+epromername);
    displaypulldown;
END;

42:BEGIN
    OpenWindow('EPROMer', 5, 5, 45, 8);
    WriteLn('----- '+epromername+' -----');
    Write('Name >');
    ReadLn(epromername);
    CloseWindow;
    displaypulldown;
END;

44:BEGIN
    ClrScr;
    WriteLn('----- Stoppen mit ^C -----');
    Exec('\'command.com', '/C readbin '+workfile+'.bin');
    taste:=ReadKey;
    displaypulldown;
END;

12:BEGIN          {change-directory}
    dirname:=getpfad('Change-Directory');
    dirchange;
    CloseWindow;
END;

51:BEGIN
    Exec('\'command.com', '/C Key-Fake "EXEC '+workfile+'.scf" 13');
    Exec('\'command.com', '/C s68c11');
    displaypulldown;
END;

52:BEGIN
    Exec('\'command.com', '/C Key-Fake "ne '+workfile+'.scf" 13');
    Exec('\'command.com', '/C '+editorname);
    displaypulldown;
END;

54:BEGIN
    ClrScr;
    WriteLn;
    WriteLn('Bitte Filenamen inkl. Extension angeben!');
    WriteLn;
    Exec('\'command.com', '/C dump');
    displaypulldown;

```

```

    END;
55:BEGIN
    ClrScr;
    Exec('\'command.com', '/C geninput');
    displaypulldown;
    END;
57:Exec('\'command.com', '/C rectos19 '+workfile);
58:Exec('\'command.com', '/C browse '+workfile+'.s19');
60:Exec('\'command.com', '/C onekey');

62:BEGIN      {environment}
    wahl:=displaymenu(7,40,6);
    CASE wahl OF
        1:soundeff:=TRUE;
        2:soundeff:=FALSE;
    END;
    removelastmenu;
END;

63:BEGIN          {directories}
    wahl:=displaymenu(6,40,6);
    IF wahl<>-1 THEN
    BEGIN
        CASE wahl OF
            1:editor:=getpfad('Editor');
            2:assembler:=getpfad('Assembler');
            3:objekt:=getpfad('Objektcode');
            4:source:=getpfad('Sourcecode');
        END;
        CloseWindow;
    END;
    removelastmenu;
END;

64:saveoptions;

71:Exec('\'command.com', '/C browse asmhc11.hlp');
72:Exec('\'command.com', '/C browse eppgm4.hlp');
73:Exec('\'command.com', '/C browse editor.hlp');
74:Exec('\'command.com', '/C browse s68c11.hlp');
75:Exec('\'command.com', '/C browse funtast.hlp');

END;
END;

```

```

PROCEDURE showtitle;
VAR taste:CHAR;
BEGIN
    OpenWindow(' ',15,8,50,13);
    WriteLn('          Assemblershell');
    WriteLn;
    WriteLn('          M.Lardelli   89');
    WriteLn;
    WriteLn('          Turbo-Pascal 4.0');
    taste:=ReadKey;
    CloseWindow;
END;

```

```
PROCEDURE options;
  VAR sound:STRING[40];
BEGIN
  {$I-}
  Reset(config);
  IF IOResult=0 THEN
  BEGIN
    ReadLn(config,workfile);
    ReadLn(config,assemblername);
    ReadLn(config,editorname);
    ReadLn(config,linkername);
    ReadLn(config,epromername);
    ReadLn(config,dirname);
    ReadLn(config,dirmaske);
    ReadLn(config,sound);
    IF sound='SOUND' THEN
      soundeff:=TRUE
    ELSE
      soundeff:=FALSE;
    Close(config);
  END
  ELSE
  BEGIN
    workfile:='tcomp';
    assemblername:='asmhc11';
    editorname:='turbo.com';
    linkername:='recbin';
    epromername:='eppgm4';
    dirname:='\tc';
    dirmaske:='*.*';
  END;
  {$I+}
  dirchange;
END;

BEGIN
  makepdmenus;
  { showtitle; }

  assign(config,'conf.asc');
  options;

  REPEAT
    choice:=scanpulldown;
    execute;
  UNTIL choice=15;
END.
```