```
      program readawi
C
C  read interpolated data of Polarstern Cruises
C
       character*30 file
       integer*4 Crunu
       REAL*4 Z(42), T(42), S(42)
C
      type*,'file name'
      accept30,file
   30 format(a30)
      open(20,file=file,status='old')
C==================================================
C     input files are in the directory OTH$daten:[socean.awi] :
C     ant2i.dat
C     ant3i.dat
C     ant5i.dat
c     ant51i.dat
C     ant7i.dat
C     ant7i.dat
C==================================================
c
  222 continue
C
      read(20,*,end=333) NSEQ ! seq number in the file
      read(20,*) Crunu ! Cruise_Number
      read(20,*) ISTAT          ! station number
      read(20,*) PHI,AMBDA  ! Latitude, Longitude (grad)
      read(20,*) NDA,MON,NYE,NHO,MIN ! day, Month, Year, Hour, Min
      read(20,*) MBDEPTH, IZLAST ! Bott_Depth (m) Max_Obs Depth (m)
      read(20,*) NUMOBS , NUMST! Number_Obs_Levels   Num_Stand_Levels
      read(20,*) MSQ ! Marsden square
C
      type*,Nseq
      type*,Crunu,ISTAT
      type*,PHI, AMBDA
      type*,NDA,MON,NYE,NHO,MIN
      type*,MBDEPTH,IZLAST
      type*,NUMOBS,NUMST
      type*,MSQ
      do 9 k=1,NUMST
      read(20,*) KK,Z(k),T(k),S(k)
      type*,KK,Z(k),T(k),S(k)
    9 continue
C=========================
      go to 222
  333 continue
      close(20)
      stop
      end
      program plotjare
C
c Maximale Feldgroessen
      parameter (maxreg=10000)
      parameter(maxx=361,MAXY=91)
c
C Definition der Variablen-Felder
c ================================
      integer*4 istyle(50),lenarr(4),ID, CRUNU
      integer*2 statnum,VFLAG
C
      CHARACTER*1 TXTARR(4),key
      CHARACTER file*40,filesn*40
      CHARACTER Ship*25,TEXT*70
C
      real
     *XG(5),YG(5)
      real xobs(40000),yobs(40000)
      real xp(maxreg),yp(maxreg)
      iundef=9999
      rundef=999.999
c Konturen der Kontinente einlesen
c ================================
      icou=0
      nreg=0
      open (2,FILE='OTH$daten:[socean.for]WORLD1.kon',status='old')
125   read(2,490,err=158) xlon,xlat
490   format(1x,2f8.3)
      icou=icou+1
      xp(icou)=xlon
      yp(icou)=xlat
      goto 125
158   close(2)
      nreg=icou
  801 format(2x,i4,a40)
      iplot=1
      isegm=1
      DATA   XMIN, XMAX, YMIN, YMAX
     *       /-180.,180.,-80.,-20./
C
C------------------
C          READ KOORDINATES
  300 continue
      open(unit=21,file='oth$daten:[socean.jare]jareall.dat'
     *,status='old')
C        I N P U T
      nstat=119
      do 333 L=1,119
      read(21,202) nseq,CRUNU,numstat,XOBS(L),YOBS(L)
      read(21,102) mmax
  102 format(2x,i3)
      do 2 k=1,mmax
      read(21,103) zz
    2 continue
  103 format(2x,f5.0,6f8.3)
  202 format(2x,3i7,2f8.2,9i7)
  333 continue
      close(21)
      type*,'number of stations =',nstat
C
C
      type*,'Type  Figure caption  (up to 70 characters)'
      TYPE*,
     *'/
     *                       /'
      accept 190,text
```

Several programs

```
  190 format(a70)
C                      OPEN  U N I R A S
      CALL GROUTE('sel mpost;ex')
      CALL GOPEN
C
C                FIRST PICTURE: STATION PLOT!!!!!!!!!!!!!!!!!!
C
         DATA LENARR /4*0/
         DATA TXTARR /4*' '/
      call gsegcr(isegm)
      xleng=230.
      yleng=95.
      CALL GWBOX(xleng,yleng,1.)
      XOFF=10.
      YOFF=50.
      CALL GVPORT(XOFF, YOFF, Xleng, Yleng)
      CALL GLIMIT(XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX)
      call GSCALE
c Definition des Gebietes 2 (Region in der geplotted wird)
         NG=5
         XG(1)=XMIN
         XG(2)=XMIN
         XG(3)=XMAX
         XG(4)=XMAX
         YG(1)=YMIN
         YG(2)=YMAX
         YG(3)=YMAX
         YG(4)=YMIN
         XG(5)=XG(1)
         YG(5)=YG(1)
C
c Laden der Regionen
         CALL GReglo(Xp,Yp,nreg,IDREG1)
         CALL GREGLO(XG,YG,NG,IDREG2)
          type*,'GREGLO DONE'
c Durchschnitt bilden
         CALL GREGOP(IDREG2,IDREG1,2,IDREG3)
      type*,'GREGOP DONE'
         IACTIV=1
         CALL GREGSS(IDREG3,IACTIV)
      type*,'GREGSS done'
      HEIGHT = 3.0
      CALL RTXFON('SWIM',1)
         IORIEN=1
         CALL GREGSO(IDREG3,IORIEN)
      TYPE*,'GREGSO DONE'
         IFILLC=-1
         FRAME=0.1
         IFRAMC=1
         IFRAMS=0
      CALL GREGDR(IDREG3,IFILLC,FRAME,IFRAMC,IFRAMS)
      TYPE*,'GREGDR DONE'
      call GSCALE
      data dbl,ntick/10.,4/
      CALL RAXTEF(4,'SWIM',1)
      CaLL RAXLFO(0,0,IUNDEF,IUNDEF)
      CALL RAXBTI(IUNDEF,RUNDEF,RUNDEF,DBL)
      CALL RAXSTI(NTICK)
      CALL RAXDIS(3,1,IUNDEF)
      CALL RAXIS2(YMIN,XMIN,HEIGHT,LENARR,TXTARR)
      CALL RAXIS2(YMAX,XMAX,HEIGHT,LENARR,TXTARR)
      type*,'AXES PLOTTED'
C   PLOT POINTS
      RD=0.3
      if(nstat.lt.6) RD = 0.45
      CALL GWICOL(RD,1)
      CALL  GDOT(xobs, yobs, nstat)
      call gsegcl(isegm)
      TYPE*, 'Stationsplot beendet'
C  Give the plot a title
C
      TPY = YMIN-5.-0.1*(YMAX-YMIN)
      TPX = 0.5*(XMIN+XMAX)
      CALL RTXFON('SWIM',1)
      CALL RTXJUS(1,3)
      CALL RTXHEI(3.0)
      CALL RTX(-1,TEXT
     * ,TPX,TPY)
C
      CALL GCLOSE
      STOP
      END
      program gortot1
      EXTERNAL err_handler
      EXTERNAL msg_handler
      include '(fsybdb)'
      character finpt*15,cmdbuf*256
      integer*4 dbproc, login, return_Code, error,id,
     * Year, month, day,NST
      real*8 Lon, Lat
      Real*4 Alon, Alat
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C
      type*,'Name of the output file'
      read(6,100) finpt
      open(unit=21,file=finpt,status='new')
  100 format(a15)
C
      call fdbfcmd(dbproc,'Execute Selgor1')
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,NST)
      call fdbnextrow(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,Id)
      call fdbbind(dbproc,2,intbind,0,Year)
      call fdbbind(dbproc,3,intbind,0,month)
      call fdbbind(dbproc,4,intbind,0,day)
      call fdbbind(dbproc,5,flt8bind,0,Lon)
      call fdbbind(dbproc,6,flt8bind,0,Lat)
      type*, NST
      do 1 j=1,NST
      call fdbnextrow(dbproc)
      Alon=sngl(Lon)
      Alat=sngl(Lat)
      type 200,j,id,Year,Month,Day,Alon,Alat
      write(21,200) J, Id, Year, Month, day,ALon, ALat
  200 format(2x,i4,1x,i6,1x,i4,1x,i2,1x,i2,1x,f7.2,1x,f7.2)
    1 Continue
      close(unit=21)
      call fdbexit()
      stop'***END***'
      end
C -----------------------------------------

C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
C      EXTERNAL        err_handler
C      EXTERNAL        msg_handler
C
      INTEGER*4       dbproc
      INTEGER*4       severity
      INTEGER*4       errno
```

```
       INTEGER*4       oserrno
       INTEGER*4       length
       INTEGER*4       return_code
C
       CHARACTER*(80)  message
C
          length = fdberrstr(errno,message)
          type *, 'DB-LIBRARY error: ', message
C
C      Check for operating system errors
C
          length = 0
          message = ' '
          length = fdboserrstr(oserrno, message)
C
          if (oserrno .ne. DBNOERR) then
             type *, 'Operating-system error: ', message
          end if
C
          return_code = fdbdead(dbproc)
C
          if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2       (severity .eq. EXSERVER)) then
             err_handler = INT_EXIT
C
          else
             err_handler = INT_CANCEL
          end if
C
          END
C
C       Message Handler
C       ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2            msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4       dbproc
      INTEGER*4       msgno
      INTEGER*4       msgstate
      INTEGER*4       severity
C
      CHARACTER*80    msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
      Program gorshipal
C-----------------------------------------------------------
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
      Integer*4  numer, dbproc, login,return_code,error
     *,nucr(1000),nucr1(1000),numstat(1000),IDmi,IDma
      Character file1*15, cmdbuf*256,ship*25,unk*7
      unk='unknown'
      type*, 'Name of output file'
      accept 101, file1
  101 format(a15)
C
      open(unit=20, file=file1,status='new')
C
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
      call fdbfcmd(dbproc,'Execute Gorshipal')
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbsetnull(dbproc,charbind,25,unk)
      call fdbbind(dbproc,1,charbind,25,Ship)
      j=0
      do while(fdbnextrow(dbproc).ne.no_more_rows)
      j=j+1
      write(20,100)j,Ship
      type 100,j,Ship
  100 format(2x,i3,2x,a25)
      end do
      close(20)
      call fdbexit()
      stop ' E N D '
      end
C -----------------------------------------

C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
C      EXTERNAL        err_handler
C      EXTERNAL        msg_handler
C
      INTEGER*4       dbproc
      INTEGER*4       severity
      INTEGER*4       errno
      INTEGER*4       oserrno
      INTEGER*4       length
      INTEGER*4       return_code
C
      CHARACTER*(80)  message
C
         length = fdberrstr(errno,message)
```

```
          type *, 'DB-LIBRARY error: ', message
C
C      Check for operating system errors
C
          length = 0
          message = ' '
          length = fdboserrstr(oserrno, message)
C
          if (oserrno .ne. DBNOERR) then
             type *, 'Operating-system error: ', message
          end if
C
          return_code = fdbdead(dbproc)
C
          if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2       (severity .eq. EXSERVER)) then
             err_handler = INT_EXIT
C
          else
             err_handler = INT_CANCEL
          end if
C
          END
C
C       Message Handler
C       ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
       INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2              msgstate,severity, msgtext)
C
       include '(fsybdb)'
C
       INTEGER*4      dbproc
       INTEGER*4      msgno
       INTEGER*4      msgstate
       INTEGER*4      severity
C
       CHARACTER*80   msgtext
         IF (MSGNO.NE.5701) THEN
C
          type *, 'DataServer message ', msgno,
     2       '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
C
         END IF
          msg_handler = DBNOSAVE
C
       END
```

```
      program gortime1
      include '(fsybdb)'
      integer*4 ID, login, dbproc,IDA(1000)
      login = fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login, 'Victor')
      dbproc = fdbopen(login, NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C     ---------------------------------------
  100 format(a15)
  111 format(2x,5i7)
C
       type*, 'Name of output file'
       read(6,100)fout
       open(unit=21, file=fout,status='new')
C
      call fdbfcmd(dbproc,'Execute Gortime ')
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,ID)
      nst=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      nst=nst+1
      write(21,111) nst,ID
      end do
      close(unit=21)
      stop '********* E N D *********'
      END
      program Gorsurv2
C    Author V.Guretsky, AWI, November 1990
C---
C     Selection of all stations within the study area of Professor Viese
C     within the Gordon subset
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
      Integer*4  dbproc, login,return_code,error,ID,BD4,SN4,MOD4,
     *YE4,MO4,DA4,Crunu,ID1(2000),N1(2000),N2(2000),N3(2000),n4(2000),
     *N5(2000),N6(2000),N7(2000)
      real*8 T8,S8,Ox8,La8,Lo8,ALO(2000),ALA(2000)
      real*4 tem(100),sal(100),oxg(100),z(100)
C
C
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
      M=0
      call fdbfcmd(dbproc,'Execute Gorsurv2')
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,ID)
      call fdbbind(dbproc,2,intbind,0,SN4)
      call fdbbind(dbproc,3,flt8bind,0,La8)
      call fdbbind(dbproc,4,flt8bind,0,Lo8)
      call fdbbind(dbproc,5,intbind,0,YE4)
      call fdbbind(dbproc,6,intbind,0,MO4)
      call fdbbind(dbproc,7,intbind,0,DA4)
      call fdbbind(dbproc,8,intbind,0,BD4)
      call fdbbind(dbproc,9,intbind,0,MOD4)
      J=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      J=J+1
      N1(J)=J
      N2(J)=SN4
      ALA(J)=LA8
      ALO(J)=LO8
      N3(J)=YE4
      N4(J)=MO4
      N5(J)=DA4
      N6(J)=BD4
      N7(J)=MOD4
      ID1(J)=ID
      end do
C
      type*,'Number of selected ID is ', J
      open(unit=21,file='Gorsurv1.dat',status='new')
      do 1 I=1,J
      type*,I
      call fdbfcmd(dbproc,'Execute Zubovsel22 %d', ID1(i))
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,BD4)
      call fdbbind(dbproc,2,flt8bind,0,T8)
      call fdbbind(dbproc,3,flt8bind,0,S8)
      call fdbbind(dbproc,4,flt8bind,0,Ox8)
      L=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      L=L+1
      z(L)=float(BD4)
      tem(L)=sngl(T8)
```

```
        sal(L)=sngl(S8)
        OXG(L)=sngl(Ox8)
        end do
C
        TTT=float(N2(I))
        if(ABS(TTT).gt.8888.) N2(I)=9999
        TTT=float(N6(I))
        if(ABS(TTT).gt.8888.) N6(I)=9999
        TTT=float(N7(I))
        if(ABS(TTT).gt.8888.) N7(I)=9999
  200   format(2x,i3,1x,i6,1x,i4,1x,f8.3,1x,f8.3,1x,i4,1x,i2,1x,i2,1x,
     *i5,1x,i5)
        write(21,200)N1(I),ID1(I),N2(I),ALA(I),ALO(I),N3(I),N4(I),
     *N5(I), N6(I), N7(I)
        write(21,200) L
        do 2 k=1,L
    2   write(21,201) z(k), tem(k), sal(k), Oxg(k)
  201   format(2x,f5.0,1x,2f8.3,1x,f6.2)
    1   continue
C
        call fdbexit()
        close(21)
        end
C ------------------------------------------
C       Error und Message Handler fuer
C       embedded SQL-Programme. In diesen mit
C       INCLUDE '(ERRMSG)' includen.
C
C       Error Handler
C       -------------
C       ERR_HANDLER - This funtion may be coded within the same program
C       or as a separate file that is compiled/linked.
C
        INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
        include '(fsybdb)'
C
C        EXTERNAL          err_handler
C        EXTERNAL          msg_handler
C
        INTEGER*4         dbproc
        INTEGER*4         severity
        INTEGER*4         errno
        INTEGER*4         oserrno
        INTEGER*4         length
        INTEGER*4         return_code
C
        CHARACTER*(80)    message
C
           length = fdberrstr(errno,message)
           type *, 'DB-LIBRARY error: ', message
C
C       Check for operating system errors
C
           length = 0
           message = ' '
           length = fdboserrstr(oserrno, message)
C
           if (oserrno .ne. DBNOERR) then
              type *, 'Operating-system error: ', message
           end if
C
           return_code = fdbdead(dbproc)
C
           if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2        (severity .eq. EXSERVER)) then
```

```
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
C      ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2            msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4       dbproc
      INTEGER*4       msgno
      INTEGER*4       msgstate
      INTEGER*4       severity
C
      CHARACTER*80    msgtext
        IF (MSGNO.NE.5701) THEN
C
        type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
C
        END IF
        msg_handler = DBNOSAVE
C
      END
      program GORSQ1
C
C         This program determines means, maximum and minimum values
C     for thestandard levels of each marsden square from Gordon subset
C         We do not use data before the "Meteor" expedition in 1924-26
C
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
C
      Integer*4  dbproc, login,return_code,error,Id(5000),
     * IDD,CN,
     * Nseq,iz4,z1,z2,number
C
      Integer*2 z(42)
C
      REAL*8 T8, S8, OX8
C
      REAL*4 T(5000), S(5000), OX(5000)
      REAL*8 Low,Loe,Lan,Las
C
      Character file1*15, cmdbuf*256,file2*15
C
      data z/0,10,20,30,50,75,100,125,150,200,250,300,350,400,
     *500,600,700,750,800,900,1000,1100,1200,1300,1400,1500,
     *1750,2000,2250,2500,2750,3000,3250,3500,3750,4000,4500,
     *5000,5500,6000,6500,7000/
C
      type*,'name of the output file'
      accept 110, file2
      open(unit=21,file=file2,status='new')
  110 format(a15)
C
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
      call fdbsetnull(dbproc,intbind,0,0)
C+++++++=================================================
C       S E L E C T I O N  O F  D A T A   F O R  T H E  S Q U A R E
      do 3 i=1,36
           Low=-180.+10.*float(i-1)
           Loe=Low+10.
CC
          do 3 j=1,5
           Lan=-30.-10.*float(j-1)
           Las=Lan-10.
      call fdbfcmd(dbproc,'Execute Square211  %f,%f,%f,%f',
     * Low,Loe,Las,Lan)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,number)
      call fdbnextrow(dbproc)
      if(NUMBER.lt.1) go to 710  ! exit to next square because of no data
C
      call fdbfcmd(dbproc,'Execute Square21  %f,%f,%f,%f',
     * Low,Loe,Las,Lan)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,IDD)
      II=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      II=II+1
      Id(II)= IDD
      end do
C
C
      IT=0    ! This is a counter
      IS=0     ! This is a counter
      IOX=0     !This is a counter
C
      do 4 jj=1, II
      IDD=ID(jj)
C      HERE FOLLOWS SELECTION OF DATA FOR THE LAYER TO ANALIZE
C
      do 4 k= 1,42
      iz4=z(k)
      call fdbfcmd(dbproc,'Execute Squaret1 %d,%d,%d',Idd,iz4)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,NN)
      callfdbnextrow(dbproc)
      if(NN.lt.1)go to 41
      call fdbfcmd(dbproc,'Execute Squaret %d,%d,%d',Idd,iz4)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,T8)
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      IT=IT+1
      T(it)=sngl(T8)
      end do
   41 continue
      call fdbfcmd(dbproc,'Execute Squares1 %d,%d,%d',Idd,iz4)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,NN)
      call fdbnextrow(dbproc)
      if(NN.lt.1) go to 42
      call fdbfcmd(dbproc,'Execute Squares %d,%d,%d',Idd,iz4)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,S8)
      do while( fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      IS=IS+1
      S(IS)=sngl(s8)
      end do
C
   42 continue
      call fdbfcmd(dbproc,'Execute Squareox1 %d,%d,%d',Idd,iz4)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,NN)
      call fdbnextrow(dbproc)
      if(NN.lt.1) go to 4
      call fdbfcmd(dbproc,'Execute Squareox %d,%d,%d',Idd,iz4)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,Ox8)
      do while( fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      IOX=IOX+1
      OX(IOX)=sngl(OX8)
      end do
C
    4 continue
C   ============END OF SELECTION FOR SQUARE AND DEPTH++++++++++
C     Get statistics for the square
C
        if(IT.gt.2) go to 50
      TMIN=0.
      TMAX=0.
      TMEAN=0.
      go to 51
   50 call STAT1(IT,T,TMIN,TMAX,TMEAN)
   51 if(IS.gt.2) go to 60
      SMIN=0.
      SMAX=0.
      SMEAN=0.
      go to 61
   60 Call STAT1(IS,S,SMIN,SMAX,SMEAN)
   61 if(IOX.gt.2) go to 70
      OXMIN=0.
      OXMAX=0.
      OXMEAN=0.
      go to 71
   70 call STAT1(IOX,Ox,OXMIN,OXMAX,OXMEAN)
   71 continue
      go to 720
  710 continue
      TMIN=0.
      TMAX=0.
      TMEAN=0.
      SMIN=0.
      SMAX=0.
      SMEAN=0.
      OXMIN=0.
      OXMAX=0.
      OXMEAN=0.
  720 continue
      write(21,111) iz4
     * ,Low,Loe,
     * Las,Lan,
     * TMIN,TMAX,TMEAN,
     * SMIN, SMAX, SMEAN,
     * OXMIN, OXMAX, OXMEAN
    3 continue
C------------
  111 format(2x, ii5,2x,4f5.0,2x,3f7.3,2x,3f7.3,2x,3f6.2)
  112 format(2x,4i5,1x,i4,1x,4i5,1x,6f8.3)
      close(21)
      call fdbexit()
      stop ' E N D '
      end
C ----------------------------------------

C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
       include '(fsybdb)'
C
C      EXTERNAL        err_handler
C      EXTERNAL        msg_handler
C
      INTEGER*4       dbproc
      INTEGER*4       severity
      INTEGER*4       errno
      INTEGER*4       oserrno
      INTEGER*4       length
      INTEGER*4        return_code
C
      CHARACTER*(80)  message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
C      ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2             msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4      dbproc
      INTEGER*4      msgno
      INTEGER*4      msgstate
      INTEGER*4      severity
C
      CHARACTER*80   msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '     severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      end
```

```
        Program Gorange
C      V,Guretsky, July, 1990,  A W I
C      Determines range of parameters at Depth from Gordons set
C-----------------------------------------------------------
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
      Integer*4  dbproc, login,return_code,error
      Character file1*15, cmdbuf*256
      type*, 'Name of output file'
      accept 100, file1
  100 format(a15)
C
  115 format(2i6)
  102 format(2x,2i7)
C
      open(unit=21, file=file1,status='new')
C
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C
      do 1 i=1,26
      nd1=200*(i-1)
      nd2=nd1+200
      call fdbfcmd(dbproc,'Execute Gomima %d,%d',nd1,nd2)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbsetnull(dbproc,intbind,0,99)
      call fdbbind(dbproc,1,flt8bind,0,Tmi8)
      call fdbbind(dbproc,2,flt8bind,0,Tma8)
      call fdbbind(dbproc,3,flt8bind,0,Smi8)
      call fdbbind(dbproc,4,flt8bind,0,Sma8)
      call fdbbind(dbproc,5,flt8bind,0,Omi8)
      call fdbbind(dbproc,6,flt8bind,0,Oma8)
      call fdbnextrow(dbproc)
      Tmi=sngl(Tmi8)
      Tma=sngl(Tma8)
      Smi=sngl(Smi8)
      Sma=sngl(Sma8)
      Omi=sngl(Omi8)
      Oma=sngl(Oma8)
C
      write(21,10)i, nd1, nd2, Tmi, Tma, Smi, Sma, Omi, Oma
    1 Continue
C
   10 format (2x,i3,i4,1x,i4,1x,f6.1,1x,f6.1,1x,f6.2,1x,f6.2,1x,f5.1,
     *1x,f5.1)
C--------------------------------------------
      close(21)
      stop ' E N D '
      end
C ---------------------------------------

C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
```

```
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
C      EXTERNAL        err_handler
C      EXTERNAL        msg_handler
C
      INTEGER*4       dbproc
      INTEGER*4       severity
      INTEGER*4       errno
      INTEGER*4       oserrno
      INTEGER*4       length
      INTEGER*4       return_code
C
      CHARACTER*(80)  message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
C      ----------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2            msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4       dbproc
      INTEGER*4       msgno
      INTEGER*4       msgstate
      INTEGER*4       severity
C
      CHARACTER*80    msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
```

```
C
      END
      program gorall2
      integer*4  ID(6400),IDD
      character`finp*15, fout*15
C     -----------------------------------
  100 format(a15)
  111 format(2x,5i7)
C
      isum=0
      do 1 i=1,4
       type*,'Name of input file'
      read(6,100)finp
      open(unit=21,file=finp,status='old')
    2 continue
      read(21,111,err=1)nst,IDD
      isum=isum+1
      id(isum)=IDD
      go to 2
    1 continue
C
      type*,'isum=',isum
C
       type*, 'Name of output file'
       read(6,100)fout
       open(unit=22, file=fout,status='new')
      type*,'Input file of all gordon Id'
      read(6,100)finp
      open(unit=21,file=finp,status='old')
C
      jsum=0
      do 3 i=1,6314
      read(21,111)nseq,IDD
C
      mark=0
      do 4 j=1,isum
    4 if(IDD.eq.id(j))mark=1
C
      if(mark.eq.0)jsum=jsum+1
      if(mark.eq.0)write(22,111) jsum,IDD
      type*,jsum
    3 continue
C
C
      close(unit=21)
      close (unit=22)
      stop '********* E N D *********'
      END
      program gorall1
      include '(fsybdb)'
      integer*4 ID, login, dbproc,IDA(1000)
      login = fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login, 'Victor')
      dbproc = fdbopen(login, NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C     ------------------------------------
  100 format(a15)
  111 format(2x,5i7)
C
       type*, 'Name of output file'
       read(6,100)fout
       open(unit=21, file=fout,status='new')
C
      call fdbfcmd(dbproc,'Execute Gorall ')
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,ID)
      nst=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      nst=nst+1
      write(21,111) nst,ID
      end do
      close(unit=21)
      stop '********* E N D *********'
      END
```

```
        PROGRAM GORDON_CONV

        CHARACTER*3200 INPUT
        CHARACTER*80 OUTPUT
        INTEGER STATUS,I,LUN,LUN1

        STATUS=LIB$GET_LUN(LUN)
        STATUS=LIB$GET_LUN(LUN1)

        OPEN(LUN,FILE='OTH$DATEN:[OZEDB.GORDON]GORDON.OLD',
        1        STATUS='OLD',RECL=3200)
        OPEN(LUN1,FILE='OTH$DATEN:[OZEDB.GORDON]GORDON.DAT',
        1        STATUS='NEW',RECORDTYPE='FIXED',RECL=80)

100     FORMAT(A3200)
110     FORMAT(A80)

10      CONTINUE
        READ(LUN,100, END=20) INPUT
        DO I=1,3200,80
                OUTPUT=INPUT(I:I+80)
                WRITE(LUN1,110) OUTPUT
        END DO
        GOTO 10
20      CONTINUE
        TYPE *,'ENDE!!'
        END
```

```
      PROGRAM DB_SIGT
C ------------------------------------------------------
C       Direktes Einlesen der Profile aus der DWB und
C       zwar mit embedded SQL ueber die stored procedure
C       'Profile '.
C       Lineare Interpolation bzgl. des Salzgehaltes und
C       des Sauerstoffes
C       und zwar mit CALL LINT(T,Z,NMAX,TOUT,IDUMMY) ETC.
C       Berechnung der pot. Temperatur und Dichte.
C       Umformattierung in PLNN-Input-Format.
C       Outputreihenfolge lautet :
C       Zaehler
C       Tiefe
C       in-situ Temperatur
C       interpolierter Salzgehalt
C       pot. Temperatur
C       Sigma Theta
C       interpolierter Sauerstoff (falls vorhanden, sonst 999.9)
C       Mit Original-Salzgehalt und  Original-Sauerstoff einbaubar.
C
C       Feb 90
C       Martin Knoche
C       Aenderungen Mar 1990
C ------------------------------------------------------
C
C       Parameter                      Beispiele
C       =========                      =========
C       IDAT    Anzahl Datenpaare      (500)
C       PID     ProfilID               (7926)
C       *TEMP   temporaere Variablen
C       *8      REAL*8 Variablen
C       N       Datenpaare pro Profil  (12)
C       Z       Tiefe                  (200.534)
C       T       Temperatur             (-1.875)
C       S       Salzgehalt             (34.379)
C       O2      Sauerstoff             (5.376)
C       PBAR    Druck in Bar           (500 bar = 5000 m)
C       PTEM    pot. Temperatur        (-1.875)
C       SIGT    pot. Dichte            (28.545)
C       LON     DB-Breite              (-54.62)
C       LAT     DB-Laenge              (23.20)
C       BDEPTH  Bodentiefe             (2650)
C       IPHI    Phi in Grad            (-74 = 74 Grad S)
C       PHI     Phi in Min             (-2.3)
C       ILAM    Lambda in Grad         (-24 = 24 Grad W)
C       LAM     Lambda in Min          (-12.9)
C       PDEPTH  Profiltiefe            (2450)
C       DATE    Datum           (FEB 22 1961 12:00AM)
C ------------------------------------------------------
      PARAMETER( IDAT = 100)
C
C     Forward declarations of the error-handler and message-handler
C     ------------------------------------------------------------
      EXTERNAL                err_handler
      EXTERNAL                msg_handler
C
       include '(fsybdb)'
C
C     Variablendeklaration
C     --------------------
      REAL
     *   Z(IDAT)
     * , T(IDAT)
     * , S(IDAT)
     * , SOUT(IDAT)
     * , TOUT(IDAT)
     * , O2(IDAT)
     * , O2OUT(IDAT)
     * , PTEM(IDAT)
     * , PBAR(IDAT)
     * , SIGT(IDAT)
     * , LON
     * , LAT
     * , PHI
     * , LAM
     * , PTTMPR
     * , ALPHA
     * , ADLPRT
C     * ,PDEPTH
C
      REAL*8
     *        LON8,
     *        LAT8,
     *        Z8TEMP,
     *        T8TEMP,
     *        S8TEMP,
     *        O8TEMP,
     *        BDEPTH8
C
      CHARACTER
     *  DS*1
     * ,FNAME*11
C     * ,cmdbuf*256
C
      INTEGER
     *  PID
     * ,ANZ
     * ,IZ(IDAT)
     * ,IDUMS
     * ,IDUMO2
C
      INTEGER*4
     *  login
     * ,dbproc
     * ,return_code
     * ,error
C
C     Array-Initialisierung
C     ---------------------
      DO 555 I=1,IDAT
         Z(I)      = 999.9
         PBAR(I)   = 999.9
         T(I)      = 999.9
         PTEM(I)   = 999.9
         S(I)      = 999.9
         SOUT(I)   = 999.9
         SIGT(I)   = 999.9
         O2(I)     = 999.9
         O2OUT(I)  = 999.9
555   CONTINUE
C
C     Install the user-supplied error-handling and
C     message-handling routines. They are defined
C     at bottom of this file
C     --------------------------------------------
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
C
C     Allocate and initialize the LOGINREC record
C     to be used to open a connection to the DataServer
C     -------------------------------------------------
      login = fdblogin()
      call fdbsetluser(login, 'MKNOCHE')
      call fdbsetlpwd(login, 'Mercy')
C
C     Oeffnen der Datenbank
C     ---------------------
      dbproc = fdbopen(login, NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C
C     Einsetzen der missing Values bei NULL
C     --------------------------------------
      call fdbsetnull(dbproc,FLT8BIND,0,999.9)
C
C
C     Setzen einiger Parameter
C     ------------------------
C     DATE=' '
C
C     Profilanzahl einlesen
C     ---------------------
      WRITE(5,110)
110   FORMAT(1X,'Profilanzahl eingeben (I3)')
      READ(6,'(I3)') ANZ
C
C     Datensatz-Abfrage
C     -----------------
      WRITE(5,120)
120   FORMAT(1X,'welcher Datensatz? Gordon (G), Aari (A) eingeben')
      READ(6,'(A1)') DS
C
C     Outputfilename sieht folgendermassen aus :
C
C     Gordon-Datensatz (G + ProfilID + Extension 001) z.B.: G7931.001
C     AARI-Datensatz   (A + ProfilID + Extension 001) z.B.: A1897.001
C     ---------------------------------------------------------------
      FNAME(1:1) = 'G'                   ! Default ist Gordon-Datensatz
      IF (DS.eq.'A'.or.DS.eq.'a') FNAME(1:1) = 'A'
      FNAME(8:11) = '.001'               ! Default-Extension
C
C     Einlesen der ProfilID's
C     -----------------------
      DO 20 I=1,ANZ
         WRITE(5,130)
130      FORMAT(1X,'ProfilID eingeben (I4)')
         READ(6,'(I6)') PID
         IUNIT     = 20 + I                    ! Outputunitnumber
         WRITE(FNAME(2:7),'(I6)') PID          ! Kernteil des Filenamens
C       ^ internes WRITE zur Typumwandlung
C
C        Oeffnen der Output-Files
C        ------------------------
         OPEN(UNIT=IUNIT,FILE=FNAME,STATUS='NEW')
C
C        Schreiben der 1.ten Kopfzeile fuer PLNN-Input
C        ---------------------------------------------
         WRITE(IUNIT,30) FNAME(1:7)
30       FORMAT(1X,'Station ',A7)
C
C        direktes Einlesen aus der DB (embedded SQL)
C        -------------------------------------------
C
C        Aufruf der stored procedure Profile
C        -----------------------------------
         call fdbfcmd(dbproc,' execute Profile %s,%d ',DS,PID)
         call fdbsqlexec(dbproc)
C
C        Uebergabe des DB-Spalteninhaltes an Programmvariablen
```

```
C        --------------------------------------------------------
          call fdbresults(dbproc)
          call fdbbind(dbproc,1,FLT8BIND,0,LON8) ! Laenge in REAL*8
          call fdbbind(dbproc,2,FLT8BIND,0,LAT8) ! Breite in REAL*8
          call fdbbind(dbproc,3,FLT8BIND,0,BDEPTH8)       ! Bodentiefe
C!!!
C        call fdbbind(dbproc,6,DATETIME,0,DATE)  ! Datum
C
          call fdbnextrow(dbproc)                ! Einlesen dieser Infozeile

C
C         Umwandlung von REAL*8 Variablen auf REAL
C         ----------------------------------------
          LON    = sngl(LON8)
          LAT    = sngl(LAT8)
          BDEPTH = sngl(BDEPTH8)
C
C         Umrechnung in Grad und Minute
C         -----------------------------
          IPHI = INT(LAT)
          PHI  = (LAT - IPHI)*60.
          ILAM = INT(LON)
          LAM  = (LON - ILAM)*60.
C
C         Information ueber die Profilposition an den Benutzer
C         ----------------------------------------------------
          WRITE(5,160)
160       FORMAT(5X,'LAT',6X,'LON',3X,'IPHI',4X,'PHI',1X,'ILAM',4X,'LAM' /)
          WRITE(5,170) LAT,LON,IPHI,PHI,ILAM,LAM
170       FORMAT(1X,2(F7.2,2X),2(1X,I4,1X,F6.1) /)
C
C         Wegschreiben in PLNN-Inputformat :
C         ----------------------------------
C
C         zweite Kopfzeile
C         ----------------
          WRITE(IUNIT,50) IPHI,PHI,ILAM,LAM       ! Positionen
C50        FORMAT(1X,2(1X,I3,1X,F5.1))
50        FORMAT(2X,2(I4,F6.1))                   ! Format aus DIST.FOR
C
C         Einlesen der Datenpaaranzahl
C         ----------------------------
          call fdbresults(dbproc)
          call fdbbind(dbproc,1,INTBIND,0,N)      ! Datenpaaranzahl id Profil
          call fdbnextrow(dbproc)          ! Abschliessen des 2.ten SELECTS der sp
C
          WRITE(5,140) PID,N
140       FORMAT(1X,'In dem Profil Nummer ',I6,' gibt es ',I4,' Datenpaare')
C
C         zeilenweises Lesen der Profildatenpaare
C         ---------------------------------------
C
C         Uebergabe des DB-Spalteninhaltes an die Programmvariablen
C         ---------------------------------------------------------
          call fdbresults(dbproc)
          call fdbbind(dbproc,1,FLT8BIND,0,Z8TEMP)          ! Tiefe (REAL*8)
          call fdbbind(dbproc,2,FLT8BIND,0,T8TEMP)          ! Temperatur (REAL*8)
          call fdbbind(dbproc,3,FLT8BIND,0,S8TEMP)          ! Salzgehalt (REAL*8)
          call fdbbind(dbproc,4,FLT8BIND,0,O8TEMP)          ! Sauerstoff (REAL*8)
C
          J = 0
          do while (fdbnextrow(dbproc).ne.NO_MORE_ROWS)
            J = J + 1                                       ! Datenpaarzahler
C
C           Umwandlung von REAL*8 Variablen auf REAL
C           ----------------------------------------
```

```
             Z(J)  = sngl(Z8TEMP)
             T(J)  = sngl(T8TEMP)
             S(J)  = sngl(S8TEMP)
             O2(J) = sngl(O8TEMP)
C
           end do                        ! Ende des Datenpaareinlesens
C
           NMAX = J                      ! Datenpaaranzahl id Profil
C
C          lineare Interpolation fuer die in-situ Temperatur
C          -----------------------------------------------------
           CALL LINT(T,Z,NMAX,TOUT,IDUMT)
           WRITE(5,43) IDUMT
43         FORMAT(1X,'Es wurden in T ',I4,' Dummywerte gefunden und
     *     linear interpoliert')
C
C          lineare Interpolation fuer den Salzgehalt
C          -----------------------------------------
           CALL LINT(S,Z,NMAX,SOUT,IDUMS)
           WRITE(5,44) IDUMS
44         FORMAT(1X,'Es wurden in S ',I4,' Dummywerte gefunden und
     *     linear interpoliert')
C
C          lineare Interpolation fuer den Sauerstoff
C          -----------------------------------------
           CALL LINT(O2,Z,NMAX,O2OUT,IDUMO2)
           WRITE(5,45) IDUMO2
45         FORMAT(1X,'Es wurden in O2 ',I4,' Dummywerte gefunden und
     *     linear interpoliert')
C
C          Berechnung der pot. Temperatur und Dichte
C          Wegschreiben in PLNN-Format und ASCII-Infofile
C          ------------------------------------------------
           DO 40 J=1,NMAX                            ! Wegschreibschleife
              PBAR(J) = Z(J)/10.                     ! Druck in Bar
              PTEM(J) = PTTMPR(SOUT(J),TOUT(J),Z(J),0.)    ! pot. Temperatur
              SIGT(J) = (1.0/ALPHA(0.0,PTEM(J),SOUT(J)))-1000.    ! pot. Dichte
C
C             umformattierter Output (=PLNN-Input) S, O2 interpoliert
C             -------------------------------------------------------
              WRITE(IUNIT,70) J,Z(J),TOUT(J),SOUT(J)
     *        ,PTEM(J),SIGT(J),O2OUT(J)
70            FORMAT(2X,I4,1X,6(F10.3))
C
C             incl Originaldaten
C             ------------------
C              WRITE(IUNIT,70) J,Z(J),T(J),SOUT(J)
C     *        ,PTEM(J),SIGT(J),O2OUT(J),S(J),O2(J)
C70            FORMAT(2X,I4,1X,8(F10.3))
C
C
40          CONTINUE                     ! Ende der Wegschreibschleife
C
            CLOSE(UNIT=IUNIT)            ! Schliessen der Profil-Outputunit
C

20       CONTINUE                        ! Ende der Profilanzahl-Schleife

C
          call fdbexit()                 ! Schliessen der DB-Library
C
         STOP 'Ende des Programmes DB_SIGT'
         END
C
C================= Subroutines und Funktions ===============
C
```

```
      SUBROUTINE LINT(XIN,Z,IANZ,XOUT,IDUMMY)
C-------------------------------------------------------------
C     Hier wird in dem Array XIN nach Dummies gesucht und dann
C     zwischen Nicht-Dummywerten linear interpoliert, der kor-
C     rigierte Output wird in dem Array XOUT an das Hauptpro-
C     gramm zurueckuebergeben, ebenso wie die Anzahl gefundener
C     Dummywerte.
C
C     Feb 90
C     Martin Knoche
C--------------------------------------------------------------
C     Parameter
C     =========
C
C     XIN         Array der Eingabevariablen
C     Z           Tiefenstufen
C     IANZ        Arraylaenge
C     XOUT        linear interpoliertes Ausgabefeld
C     IDUMMY      Anzahl der gefundenen Dummy-Werte
C     IFLAG       Null, falls kein Dummy gefunden wurde, sonst 1
C                 wird nach jeder Interpolation neu initialisiert
C--------------------------------------------------------------
C
C     Varaiablendeklaration
C     ---------------------
      REAL XIN(IANZ),XOUT(IANZ),Z(IANZ)
      INTEGER IFLAG,IDUMMY
C
C     Setzen von Parametern
C     ---------------------
      IDUMMY  = 0
      IFLAG   = 0
      L       = 0                         ! Dummy-Zaehler
      K       = 0                         ! Dummy-Index
C
C     falls Start- und Stopwerte = Dummy
C     ---------------------------------
      IF ( XIN(1)    .eq.999.9 ) XIN(1) = XIN(2)          ! 1.ter Wert
      IF ( XIN(IANZ).eq.999.9 ) XIN(I) = XIN(I-1)         ! letzter Wert
C
C     Datenschleife
C     -------------
      DO 10 I=1,IANZ
        XOUT(I) = XIN(I)
        IF ( XOUT(I).eq.999.9 ) THEN
           L = L + 1                      ! Dummy-Zaehler
           IF ( IFLAG.eq.1 ) GOTO 20      ! Suche des naechsten nicht-Dummywertes
              K      = I                  ! Dummy-Index
              X1     = XOUT(I-1)          ! letzter nicht-Dummywert
              Z1     = Z(I-1)             ! zugehoeriger Tiefenwert
              IFLAG = 1                   ! Dummy gefunden -> IFLAG = 1
              GOTO 20                     ! Suche des naechsten nicht-Dummywertes
        ELSE                      ! XOUT ungleich Dummy
           IF ( IFLAG.eq.0 ) GOTO 20      ! Suche des naechsten nicht-Dummywertes
              X2     = XOUT(I)
              Z2     = Z(I)
              SLOPE = (X2-X1)/(Z2-Z1)     ! Steigung der Fitgeraden
              X0     = X1 - SLOPE*Z1      ! X-Achsenabschnitt
C
C             lineare Interpolation zwischen Nicht-Dummywerten
C             ------------------------------------------------
              DO 30 J=K,I-1               ! Interpolationsschleife
                 XOUT(J) = X0 + SLOPE*Z(J)      ! Interpolation
30            CONTINUE
           IFLAG = 0              ! IFLAG-Initalisierung fuer naechste Dummysuche
        END IF                    ! Ende der Dummy-Abfrage
```

```
20    CONTINUE                       ! schlichte Sprungadresse
10    CONTINUE                       ! Ende der Daten-Schleife
C
      IDUMMY = L                             ! Anzahl gefundener Dummies
C
      RETURN
      END
C
C
C ---------------------------------------------------------------------
      FUNCTION PTTMPR ( SALZ, TEMP, PRES, RFPRES )
C ---------------------------------------------------------------------
C Checkwert: PTTMPR = 36.89073 DegC
C       fuer SALZ   =    40.0 psu
C            TEMP   =    40.0 DegC
C            PRES   = 10000.000 dbar
C            RFPRES =     0.000 dbar
C
      PARAMETER ( CT2  = 0.29289322,  CT3  =  1.707106781,
     1            CQ2A = 0.58578644,  CQ2B =  0.121320344,
     2            CQ3A = 3.414213562, CQ3B = -4.121320344 )
C
      P  = PRES
      T  = TEMP
      DP = RFPRES-PRES
      DT = DP*ADLPRT ( SALZ, T, P )
      T  = T + 0.5*DT
      Q  = DT
      P  = P + 0.5*DP
      DT = DP*ADLPRT ( SALZ, T, P )
      T  = T + CT2*(DT-Q)
      Q  = CQ2A*DT + CQ2B*Q
      DT = DP*ADLPRT ( SALZ, T, P )
      T  = T + CT3*(DT-Q)
      Q  = CQ3A*DT + CQ3B*Q
      P  = RFPRES
      DT = DP*ADLPRT ( SALZ, T, P )
      PTTMPR = T + (DT-Q-Q)/6.0
      END
C
C
C ---------------------------------------------------------------------
      FUNCTION ADLPRT ( SALZ, TEMP, PRES )
C ---------------------------------------------------------------------
C Berechnet aus dem Salzgehalt/psu (SALZ), der in-situ Temperatur/degC
C (TEMP) und dem in-situ Druck/dbar (PRES) den adiabatischen Temperatur-
C gradienten/(K Dbar^-1) ADLPRT.
C Checkwert: ADLPRT =     3.255976E-4 K dbar^-1
C       fuer SALZ   =    40.0 psu
C            TEMP   =    40.0 DegC
C            PRES   = 10000.000 dbar
C
      PARAMETER ( S0 = 35.0,
     1 A0 =  3.5803E-5,  A1 =  8.5258E-6,  A2 = -6.8360E-8,
     2 A3 =  6.6228E-10, B0 =  1.8932E-6,  B1 = -4.2393E-8,
     3 C0 =  1.8741E-8,  C1 = -6.7795E-10, C2 =  8.7330E-12,
     4 C3 = -5.4481E-14, D0 = -1.1351E-10, D1 =  2.7759E-12,
     5 E0 = -4.6206E-13, E1 =  1.8676E-14, E2 = -2.1687E-16 )
C
      DS = SALZ-S0
      ADLPRT = ( ( (E2*TEMP + E1)*TEMP + E0 )*PRES
     1          + ( (D1*TEMP + D0)*DS
     2             + ( (C3*TEMP + C2)*TEMP + C1 )*TEMP + C0 ) )*PRES
     3  + (B1*TEMP + B0)*DS +  ( (A3*TEMP + A2)*TEMP + A1 )*TEMP + A0
      END
C
```

```
C
C ---------------------------------------------------------------------
      FUNCTION ALPHA(P,T,S)
C ---------------------------------------------------------------------
C     EQUATION OF STATE FOR SEAWATER PROPOSED BY JPOTS 1980
C       UNITS:
C             PRESSURE        P          BARS
C             TEMPERATURE     T          DEG CELCIUS (IPTS-68)
C             SALINITY        S          NSU (IPSS-78)
C             DENSITY         RHO        KG/M**3
C             SPEC. VOL.      ALPHA      M**3/KG
C     CHECK VALUE:
C             ALPHA = 9.435561E-4 M**3/KG
C             FOR:
C                     S = 40 NSU
C                     T = 40 DEG C
C                     P = 1000 BARS
C PDP11 GETESTET: 0.94355614 E-03
C END OF DOC
      IMPLICIT INTEGER*2 (I-N)
      REAL P,T,S,RHO,SR,R1,R2,R3,R4
      REAL A,B,C,D,E,A1,B1,AW,BW,K,KO,KW
      EQUIVALENCE (E,D,B1,R4),(BW,B,R3),(C,A1,R2)
      EQUIVALENCE (AW,A,R1,RO),(KW,KO,K)
      SR=SQRT(ABS(S))
C PURE WATER DENSITY AT ATM PRESS.
      R1=((((6.536332E-9*T-1.120083E-6)*T+1.001685E-4)*T
     *-9.095290E-3)*T+6.793952E-2)*T+999.842594
C SEAWATER DENSITY AT ATM PRESS.
      R2=(((5.3875E-9*T-8.2467E-7)*T+7.6438E-5)*T-4.0899E-3)*T
     *+8.24493E-1
      R3=(-1.6546E-6*T+1.0227E-4)*T-5.72466E-3
      R4=4.8314E-4
      RHO=(R4*S + R3*SR + R2)*S + R1
C SPECIFIC VOL. AT ATM PRESS
      ALPHA=1.0/RHO
      IF(P.EQ.0.0) RETURN
C COMPUTE SECANT BULK MODULUS K(P,T,S)
      E=(9.1697E-10*T+2.0816E-8)*T-9.9348E-7
      BW=(5.2787E-8*T-6.12293E-6)*T+8.50935E-5
      B=BW + E*S
C
      D=1.91075E-4
      C=(-1.6078E-6*T-1.0981E-5)*T+2.2838E-3
      AW=((-5.77905E-7*T+1.16092E-4)*T+1.43713E-3)*T
     *+3.239908
      A=(D*SR + C)*S + AW
C
      B1=(-5.3009E-4*T+1.6483E-2)*T+7.944E-2
      A1=((-6.1670E-5*T+1.09987E-2)*T-0.603459)*T+54.6746
      KW=(((-5.155288E-5*T+1.360477E-2)*T-2.327105)*T
     *+148.4206)*T+19652.21
C COMPUTE K(0,T,S)
      KO=(B1*SR + A1)*S + KW
C EVALUATE K(P,T,S)
      K=(B*P + A)*P + KO
      ALPHA=ALPHA*(1.0-P/K)
      RETURN
      END
C
C ----------------------------------------
C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
```

```
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
C      EXTERNAL          err_handler
C      EXTERNAL          msg_handler
C
      INTEGER*4         dbproc
      INTEGER*4         severity
      INTEGER*4         errno
      INTEGER*4         oserrno
      INTEGER*4         length
      INTEGER*4         return_code
C
      CHARACTER*(80)    message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
C      ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2           msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4         dbproc
      INTEGER*4         msgno
      INTEGER*4         msgstate
      INTEGER*4         severity
C
      CHARACTER*80      msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
```

```
C
          END IF
           msg_handler = DBNOSAVE
C
        END
C
```

```
      Program Deldaba2
C     This program reads  Id of stations to be deleted
C          and then delete them
C      V.Guretsky, AWI, May, 1990
C----------------------------------------------------------
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
C
      Integer*4  ncount, dbproc, login,return_code,error,id,nc
C
      Character file1*15, cmdbuf*256, ship*15, file2*15, Date*20
C               -------I N P U T------
           type*, 'Name of intput file'
      accept 100, file1
  100 format(a15)
      open(unit=20, file=file1,status='old')
           type*,'Name of output file for the protocol of deleation'
      accept 100, file2
      open(unit=21,file=file2,status='new')
           Type*,'Insert Date_Time  of transaction as Character*20'
      accept 101, Date
  101 format(A20)
C     ----------------------------------------
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C
      write(21,201)
      write(21,202) Date
  201 format(2x,' PROTOCOL OF DELETION OF STATIONS WITHIN SoOceanDB')
  202 format(15x, a20)
      i=0
  113 continue
      read(20,200, end=112) ncount, Id
      call fdbfcmd(dbproc,'Execute Delete1 %d', Id)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      i=i+1
      type 200, i,Id, Nc, Ship
      write(21,200) i, Id, Nc, Ship
      go to 113
  112 continue
      close(21)
      close(20)
  200 format(2X, 3i7, 2x, a15)
      call fdbexit()
      stop ' E N D '
      end
C ----------------------------------------
C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
```

```
C      EXTERNAL         err_handler
C      EXTERNAL         msg_handler
C
      INTEGER*4         dbproc
      INTEGER*4         severity
      INTEGER*4         errno
      INTEGER*4         oserrno
      INTEGER*4         length
      INTEGER*4         return_code
C
      CHARACTER*(80)    message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
C      ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2            msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4        dbproc
      INTEGER*4        msgno
      INTEGER*4        msgstate
      INTEGER*4        severity
C
      CHARACTER*80     msgtext
        IF (MSGNO.NE.5701) THEN
C
        type *, 'DataServer message ', msgno,
     2     '   state ', msgstate, '     severity ',
     3     severity,' ', msgtext
C
        END IF
        msg_handler = DBNOSAVE
C
      END
      Program Deldaba1
C     This program reads  Id of stations to be deleted
C          and then delete them
C      V.Guretsky, AWI, May, 1990
C-------------------------------------------------------------
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
C
      Integer*4  ncount, dbproc, login,return_code,error,id,nc
C
      Character file1*15, cmdbuf*256, ship*15, file2*15, Date*20
C               -------I N P U T------
           type*, 'Name of intput file'
      accept 100, file1
  100 format(a15)
      open(unit=20, file=file1,status='old')
           type*,'Name of output file for the protocol of deleation'
      accept 100, file2
      open(unit=21,file=file2,status='new')
           Type*,'Insert Date_Time  of transaction as Character*20'
      accept 101, Date
  101 format(A20)
C     -----------------------------------------
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C
      write(21,201)
      write(21,202) Date
  201 format(2x,' PROTOCOL OF DELETION OF STATIONS WITHIN SoOceanDB')
  202 format(15x, a20)
      i=0
  113 continue
      read(20,200, end=112) ncount, Id, Nc, Ship
      call fdbfcmd(dbproc,'Execute Delete1 %d', Id)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      i=i+1
      type 200, i,Id, Nc, Ship
      write(21,200) i, Id, Nc, Ship
      go to 113
  112 continue
      close(21)
      close(20)
  200 format(2X, 3i7, 2x, a15)
      call fdbexit()
      stop ' E N D '
      end
C -----------------------------------------
C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
```

```
C       EXTERNAL          err_handler
C       EXTERNAL          msg_handler
C
      INTEGER*4         dbproc
      INTEGER*4         severity
      INTEGER*4         errno
      INTEGER*4         oserrno
      INTEGER*4         length
      INTEGER*4         return_code
C
      CHARACTER*(80)   message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
C      ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2             msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4        dbproc
      INTEGER*4        msgno
      INTEGER*4        msgstate
      INTEGER*4        severity
C
      CHARACTER*80    msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '     severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
```

```
                 Program differ1
C     This program searches for stations which are not duplicates
C     within the tables of possible duplicate stations
C            V.Guretsky, May, 1990, AWI
C     -----------------------------------
      real lon(2), lat(2), z(50), s(50,2),O2(50,2),t(50,2),
     * depth(2), modepth(2), dt(50),ds(50),dox(50),
     * sigt(50,2), sigpot(50,2), pbar(50), tpot(50,2), dsig(50,2),
     * dtp(50,2), dtdt(50,2),
     * sr(50),tr(50),Or(50),lonr,latr,modepthr, sig0(50,2)
C
      integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
      character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15, x*1

      integer*4 nc(2), id(2), ncr, idr
C -----------------------------------------------------------------
      ncount=0
C           I N P U T
      type*, 'Name of input file'
      accept 100, file1
  100 format(a12)
      open(unit=21, file=file1,status='old')
      type *, 'Name of outputfile for the numbers of nonduplicates'
      accept 100, file2
      open(unit=22,file=file2,status='new')
  555 continue
      read(21,111,end=112) nnn
      read(21,111) id
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
   52 format(2x,2f7.0)
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55)z(k),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2)
      type*,nnn,n
C        CHECK IF THIS PAIR HAS ALREADY BEEN PROCESSED
      if(depth(1).lt.0..and.depth(2).lt.0.) goto 555
  111 format(2x,5i7)
C      ----------------------------------------------------
C       Coordinates   Criterium for duplicates
      if(abs(dlon).ge.0.1) go to 555
    2 if(abs(dlat).ge.0.1) go to 555
C   ---------------------------------------------------
C       here check equality of month and day
      if(nmonth(1).eq.nmonth(2).and.nday(1).eq.nday(2))go to 555
C      ----------------------------------------------------
C            T Y P E   S T A T I O N S   O N   T H E   S C R E E N
  444 continue
      type 111, nnn
      type 111, id
      type 50, nc(1),Ship1, nc(2),ship2
      type 111, numst
      type 51, Lon, dlon
      type 51, Lat, dlat
      type 52, Depth
      type 52, Modepth
      type 111, nyear
      type 111, nmonth
      type 111, nday
      type 111, nhour
      type 111, nob
      type 111, nms
      type 111, n
C     ----------------------------
   56 format(a1)
C     ----------------------------
      do 28 k = 1, n
   28 type 55, z(k), (t(k,j),j=1,2), dt(k), (s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k)
C     ----------------------------
      type*,'$$$$$  type station again? 0 - no  1 - yes'
      accept 57,k
      if(k)445,445,444
   57 format(2i1)
  445 continue
      type*,'$$$$$  TYPE:  duplicates 0  different  1'
      accept 57, k
      if(k) 555,555,557
  557 continue
C     -------------------------------------------
      ncount=ncount+1
C     ---------------------------------------------------
C                         O U T P U T
      write(22,200) ncount,id(1),nc(1), ship1
  200 format(2x,3i7,2x,a15)
      goto555
  112 continue
      close(unit=21)
      close(unit=22)
      stop '*** E N D ***'
      end
```

```
        program Aargor9
C       This program select data from the Standard_data table
C       for the specified Gordon Station and one or more AARI stations.
C       After interpolation to the standard depths Gordon station is
C       compared with AARI station(s). IF THERE IS COINCIDENCE
C       only on less than 10  percents of standard levels for Gordon and
C       Aari Station Aari-Station_Id# is written together with
C       Gordon_Station_Id# in to the output file
C
C       V.Guretsky, AWI, June 1990
        include '(fsybdb)'
        integer*4 IDG, IDA, login, dbproc,IDAR(1000),CRU(1000),Crunum,
      * CRUFIN(1000), IDAFIN(1000)
C
         character file1*15, file2*15
C
        real*8 LOGOR8,LOAAR8,LAGOR8,LAAAR8,BDGOR8,BDAAR8,MOGOR8,MOAAR8
        real*8 T8,O8,S8,Z8
        real*4 tema(42), sala(42),oxya(42),temg(42),salg(42),oxyg(42),
      *        zg1(80),tg1(80),sg1(80),og1(80),zst(42),za1(42),
      *        fob1(80), zob1(80) ,
      *        dt(80),ds(80),dx(80)
C
        login = fdblogin()
        call fdbsetluser(login,'SOCEAN')
        call fdbsetlpwd(login, 'Victor')
        dbproc = fdbopen(login, NULL)
        call fdbuse(dbproc,'SouthernOceanDB')
        data zst /0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,
      * 250.,300.,350.,400.,500.,600.,700.,750.,800.,900.,
      * 1000.,1100.,1200.,1300.,1400.,1500.,1750.,2000.,2250.,2500.,
      * 2750.,3000.,3250.,3500.,3750.,4000.,4500.,5000.,5500.,6000.,
      * 6500.,7000./
C
C       ------------------------------------
  100 format(a15)
  111 format(2x,10i7)
C
         type*, 'Name of input file'
         read(6,100)file1
         open(unit=21, file=file1,status='old')
         type*, 'Name of output file'
         read(6,100)file2
         open(unit=22, file=file2,status='new')
C
        iaar=0
        iseq=0
  222 continue
        LOOP=0
        read(21,111,end=333,err=222) nseq, IDG, nst,(IDAR(i),cru(i),
      * i=1,nst)
        iaar=iaar+nst
        type*,nseq
C
C   Selection of standard data for the gordon data
C
        call fdbsetnull(dbproc,flt8bind,0,99.)
        call fdbfcmd(dbproc,'Execute Stadata %d', IDG)
        call fdbsqlexec(dbproc)
        call fdbresults(dbproc)
        call fdbbind(dbproc,1,flt8bind,0,Z8)
        call fdbbind(dbproc,2,flt8bind,0,T8)
        call fdbbind(dbproc,3,flt8bind,0,S8)
        call fdbbind(dbproc,4,flt8bind,0,O8)
        m=0
        do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      m=m+1
      zg1(m)=sngl(Z8)
      tg1(m)=sngl(T8)
      sg1(m)=sngl(S8)
      Og1(m)=sngl(O8)
      end do
C
      if(m.eq.0) go to 222
C
      do 11 k=1,42
      temg(k)=0.
      salg(k)=0.
      oxyg(k)=0.
   11 continue
C
C     INTERPOLATION OF GORDON DATA
C
C     I N T E R P O L A T I O N
      fmin=-2.3
      fmax=29.
      mt=inter(m, zg1, tg1, fmin, fmax, temg, zst, nob2, fob1, zob1)
      fmin=27.
      fmax=36.2
      ms=inter(m, zg1, sg1, fmin, fmax, salg, zst, nob2, fob1, zob1)
      fmin=1.
      fmax=14.
      mox=inter(m, zg1, og1, fmin, fmax, oxyg, zst, nob2, fob1, zob1)
C
      mmax=max0(mt,ms,mox)
C
C

C       AARI STATIONS LOOP
      loop=0
      do 4 j = 1, nst
C
      do 12 k=1,42
      tema(k)=0.
      sala(k)=0.
      oxya(k)=0.
   12 continue
C
C
      IDA=IDAR(j)
      call fdbfcmd(dbproc,'Execute Stadata %d', IDA)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,Z8)
      call fdbbind(dbproc,2,flt8bind,0,T8)
      call fdbbind(dbproc,3,flt8bind,0,S8)
      call fdbbind(dbproc,4,flt8bind,0,O8)
      m=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      m=m+1
      za1(m)=sngl(Z8)
      tema(m)=sngl(T8)
      sala(m)=sngl(S8)
      Oxya(m)=sngl(O8)
      end do
C
      if(m.eq.0) go to 4
C
C.        here we make rearrangement of aari station
      L=0
      do 92 k=1,42
      L=L+1
   92 if(za1(1).eq.0.) go to 93
   93 k1=m+L
      k2=m+1
      do 94 k=1,m
      za1(k1-k)=za1(k2-k)
      tema(k1-k)=tema(k2-k)
      sala(k1-k)=sala(k2-k)
   94 oxya(k1-k)=oxya(k2-k)
      do 95 k=1,L
      za1(k)=99.
      tema(k)=99.
      sala(k)=99.
      oxya(k)=99.
   95 continue
C
C
      nnn=max0(m,mmax)
      do 5 k=1,nnn
      dt(k)=temg(k)-tema(k)
      ds(k)=salg(k)-sala(k)
      dx(k)=oxyg(k)-oxya(k)
    5 continue
C
C        HERE MAKE COMPARISON OF GORDON AND AARI STANDARD DATA
      mt=0
      ms=0
      mmm=nnn
         do 22 k=1,nnn
      if(temg(k).lt.-2.3.or.temg(k).gt.29.) go to 23
      if(salg(k).lt.25..or.salg(k).gt.36.3) go to 23
      if(tema(k).lt.-2.3.or.tema(k).gt.29.) go to 23
      if(sala(k).lt.25..or.sala(k).gt.36.3) go to 23
      if(abs(dt(k)).lt.0.005) mt=mt+1
      if(abs(ds(k)).lt.0.005) ms=ms+1
      go to 22
   23 mmm=mmm-1
   22 continue
C
      if(mmm.eq.0) go to 222
      mtp=mt*100/mmm
      msp=ms*100/mmm
      if(mtp.ge.10) go to 4
      if(msp.ge.10) go to 4
      LOOP=LOOP+1
      IDAFIN(LOOP)= IDA
      CRUFIN(LOOP)= CRU(j)
    4 continue
C
C        HERE MAKE COMPARISON OF GC
C       HERE WE WRITE INFORMATION FOR STATIONS
   44 continue
      if(LOOP)222,222,46
CC    45 write(22,111) nseq, IDG, LOOP
CC       type 111,nseq,IDG,LOOP
CC       go to 222
   46 iseq=iseq+1
      write(22,111) iseq, IDG, LOOP,(IDAFIN(j),CRUFIN(j),j=1,LOOP)
      type 111, iseq, IDG,LOOP,(IDAFIN(j),CRUFIN(j),j=1,LOOP)
      go to 222
  333 continue
      mp=iseq*100/nseq
      type*,' 80-percent ratio =',mp
      type*,'total aari =',iaar
      type*,'total gordon =',nseq
      close(unit=21)
      close(unit=22)
```

```
call fdbexit()
stop '********* E N D *********'
END
```

SOcean 4-FOR

```
      program Aargor3
C     This program select data from the Standard_data table
C     for the specified Gordon Station and one or more AARI stations.
C     After interpolation to the standard depths Gordon station is
C     compared with AARI station(s). IF THERE IS COINCIDENCE
C     on  at least 80 percents of standard levels for Gordon and
C     Aari Station Aari-Station_Id# is written together with
C     Gordon_Station_Id# in to the output file
C
C     V.Guretsky, AWI, June 1990
      include '(fsybdb)'
      integer*4 IDG, IDA, login, dbproc,IDAR(1000),CRU(1000),Crunum,
     * CRUFIN(1000), IDAFIN(1000)
C
       character file1*15, file2*15
C
      real*8 LOGOR8,LOAAR8,LAGOR8,LAAAR8,BDGOR8,BDAAR8,MOGOR8,MOAAR8
      real*8 T8,O8,S8,Z8
      real*4 tema(42), sala(42),oxya(42),temg(42),salg(42),oxyg(42),
     *        zg1(80),tg1(80),sg1(80),og1(80),zst(42),za1(42),
     *        fob1(80), zob1(80) ,
     *        dt(80),ds(80),dx(80)
C
      login = fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login, 'Victor')
      dbproc = fdbopen(login, NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
      data zst /0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,
     * 250.,300.,350.,400.,500.,600.,700.,750.,800.,900.,
     * 1000.,1100.,1200.,1300.,1400.,1500.,1750.,2000.,2250.,2500.,
     * 2750.,3000.,3250.,3500.,3750.,4000.,4500.,5000.,5500.,6000.,
     * 6500.,7000./
C
C     ------------------------------------
  100 format(a15)
  111 format(2x,10i7)
C
       type*, 'Name of input file'
       read(6,100)file1
       open(unit=21, file=file1,status='old')
       type*, 'Name of output file'
       read(6,100)file2
       open(unit=22, file=file2,status='new')
C
      iaar=0
      iseq=0
  222 continue
      LOOP=0
      read(21,111,end=333,err=222) nseq, IDG, nst,(IDAR(i),cru(i),
     * i=1,nst)
      iaar=iaar+nst
C
C  Selection of standard data for the gordon data
C
      call fdbsetnull(dbproc,flt8bind,0,99.)
      call fdbfcmd(dbproc,'Execute Stadata %d', IDG)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,Z8)
      call fdbbind(dbproc,2,flt8bind,0,T8)
      call fdbbind(dbproc,3,flt8bind,0,S8)
      call fdbbind(dbproc,4,flt8bind,0,O8)
      m=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      m=m+1
      zg1(m)=sngl(Z8)
      tg1(m)=sngl(T8)
      sg1(m)=sngl(S8)
      Og1(m)=sngl(O8)
      end do
C
      if(m.eq.0) go to 222
C
      do 11 k=1,42
      temg(k)=0.
      salg(k)=0.
      oxyg(k)=0.
   11 continue
C
C     INTERPOLATION OF GORDON DATA
C
C     I N T E R P O L A T I O N
      fmin=-2.3
      fmax=29.
      mt=inter(m, zg1, tg1, fmin, fmax, temg, zst, nob2, fob1, zob1)
      fmin=27.
      fmax=36.2
      ms=inter(m, zg1, sg1, fmin, fmax, salg, zst, nob2, fob1, zob1)
      fmin=1.
      fmax=14.
      mox=inter(m, zg1, og1, fmin, fmax, oxyg, zst, nob2, fob1, zob1)
C
      mmax=max0(mt,ms,mox)
C
C

C       AARI STATIONS LOOP
      loop=0
      do 4 j = 1, nst
C
      do 12 k=1,42
      tema(k)=0.
      sala(k)=0.
      oxya(k)=0.
   12 continue
C
C
      IDA=IDAR(j)
      call fdbfcmd(dbproc,'Execute Stadata %d', IDA)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,Z8)
      call fdbbind(dbproc,2,flt8bind,0,T8)
      call fdbbind(dbproc,3,flt8bind,0,S8)
      call fdbbind(dbproc,4,flt8bind,0,O8)
      m=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      m=m+1
      za1(m)=sngl(Z8)
      tema(m)=sngl(T8)
      sala(m)=sngl(S8)
      Oxya(m)=sngl(O8)
      end do
C
      if(m.eq.0) go to 4
C
C.         here we make rearrangement of aari station
      L=0
      do 92 k=1,42
      L=L+1
   92 if(za1(1).eq.0.) go to 93
   93 k1=m+L
      k2=m+1
      do 94 k=1,m
      zal(k1-k)=zal(k2-k)
      tema(k1-k)=tema(k2-k)
      sala(k1-k)=sala(k2-k)
   94 oxya(k1-k)=oxya(k2-k)
      do 95 k=1,L
      zal(k)=99.
      tema(k)=99.
      sala(k)=99.
      oxya(k)=99.
   95 continue
C
C
      nnn=max0(m,mmax)
      do 5 k=1,nnn
      dt(k)=temg(k)-tema(k)
      ds(k)=salg(k)-sala(k)
      dx(k)=oxyg(k)-oxya(k)
    5 continue
C
C        HERE MAKE COMPARISON OF GORDON AND AARI STANDARD DATA
      mt=0
      ms=0
      mmm=nnn
         do 22 k=1,nnn
      if(temg(k).lt.-2.3.or.temg(k).gt.29.) go to 23
      if(salg(k).lt.25..or.salg(k).gt.36.3) go to 23
      if(tema(k).lt.-2.3.or.tema(k).gt.29.) go to 23
      if(sala(k).lt.25..or.sala(k).gt.36.3) go to 23
      if(abs(dt(k)).lt.0.005) mt=mt+1
      if(abs(ds(k)).lt.0.005) ms=ms+1
      go to 22
   23 mmm=mmm-1
   22 continue
C
      if(mmm.eq.0) go to 222
      mtp=mt*100/mmm
      msp=ms*100/mmm
      if(mtp.ge.80.and.msp.ge.80)go to 48
      go to 4
   48 LOOP=LOOP+1
      IDAFIN(LOOP)= IDA
      CRUFIN(LOOP)= CRU(j)
    4 continue
C
C        HERE MAKE COMPARISON OF GC
C       HERE WE WRITE INFORMATION FOR STATIONS
   44 continue
      if(LOOP)222,222,46
CC   45 write(22,111) nseq, IDG, LOOP
CC      type 111,nseq,IDG,LOOP
CC      go to 222
   46 iseq=iseq+1
      write(22,111) iseq, IDG, LOOP,(IDAFIN(j),CRUFIN(j),j=1,LOOP)
      type 111, iseq, IDG,LOOP,(IDAFIN(j),CRUFIN(j),j=1,LOOP)
      go to 222
  333 continue
      mp=iseq*100/nseq
      type*,' 80-percent ratio =',mp
      type*,'total aari =',iaar
      type*,'total gordon =',nseq
      close(unit=21)
      close(unit=22)
      call fdbexit()
```

```
          Program Checkrng2
C        V,Guretsky, July, 1990,   A W I
C        Check range of parameters for each station
C------------------------------------------------------------
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
C
      Integer*4  dbproc, login,return_code,error,Id1,Id2,Idsel,
     *K, J, IDALL(35000),z(100),depth,nd1(26),nd2(26),is,ncount
C
      real*8 t8,s8,O8
C
      real*4
     *t(100),s(100),Ox(100),tmi(26),tma(26),
     *smi(26),sma(26),Omi(26),Oma(26)
C
      Character file1*15, cmdbuf*256,file2*15
C
      type*,'min and max Id as 2i6'
      accept 115, id1,id2
C
c
      type*, 'Name of output file'
      accept 110, file1
  110 format(a15)
C
  115 format(2i6)
C
      open(unit=21, file=file1,status='new')
C
      mt=1
      ms=2
      mo=3
      ncount=0
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C
      call fdbfcmd(dbproc,'Execute Sel01 %d,%d',Id1,Id2)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,idsel)
      J=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      J=J+1
      IDALL(J)=idsel
      end do
C-----------------------------------------------------------
      call fdbsetnull(dbproc,flt8bind,0,-12.)
      do1 is =1, J
      idsel=idall(is)
      call fdbfcmd(dbproc,'Execute Sel02 %d',idsel)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1, intbind,0,depth)
      call fdbbind(dbproc,2,flt8bind,0,T8)
      call fdbbind(dbproc,3,flt8bind,0,S8)
      call fdbbind(dbproc,4,flt8bind,0,O8)
      K=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      K=K+1
```

```
         z(k)=depth
         t(k)=sngl(T8)
         s(k)=sngl(S8)
CC         Ox(k)=sngl(O8)
         end do
         nz=k
C
         do 2 k=1,nz
         if(t(k).eq.-12.)go to 24
         go to 25
      24 ncount=ncount+1
         write(21,100)IDALL(is),mt,z(k),t(k)
      25 if(s(k).eq.-12.)go to 23
         go to 2
      23 ncount=ncount+1
         write(21,100) IDALL(is), ms,z(k),s(k)
       2  continue
C
    107 format(2x,i5)
    100 format(2x,i6,1x,i1,1x,i4,1x,f8.3)
C-----------------------------------------------
       1 continue
C
        close(21)
        stop ' E N D '
        end
C ------------------------------------------

C       Error und Message Handler fuer
C       embedded SQL-Programme. In diesen mit
C       INCLUDE '(ERRMSG)' includen.
C
C       Error Handler
C       -------------
C       ERR_HANDLER - This funtion may be coded within the same program
C       or as a separate file that is compiled/linked.
C
        INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
        include '(fsybdb)'
C
C        EXTERNAL         err_handler
C        EXTERNAL         msg_handler
C
        INTEGER*4        dbproc
        INTEGER*4        severity
        INTEGER*4        errno
        INTEGER*4        oserrno
        INTEGER*4        length
        INTEGER*4        return_code
C
        CHARACTER*(80)   message
C
           length = fdberrstr(errno,message)
           type *, 'DB-LIBRARY error: ', message
C
C       Check for operating system errors
C
           length = 0
           message = ' '
           length = fdboserrstr(oserrno, message)
C
           if (oserrno .ne. DBNOERR) then
              type *, 'Operating-system error: ', message
           end if
C
```

```
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C     Message Handler
C     ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2            msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4      dbproc
      INTEGER*4      msgno
      INTEGER*4      msgstate
      INTEGER*4      severity
C
      CHARACTER*80   msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
```

```
         Program Checkrng1
C       V,Guretsky, July, 1990,  A W I
C       Check range of parameters for each station
C------------------------------------------------------------
       EXTERNAL err_handler
       External msg_handler
       include '(fsybdb) '
C
       Integer*4  dbproc, login,return_code,error,Id1,Id2,Idsel,
      *K, J, IDALL(35000),z(100),depth,nd1(26),nd2(26),is,ncount
C
       real*8 t8,s8,O8
C
       real*4
      *t(100),s(100),Ox(100),tmi(26),tma(26),
      *smi(26),sma(26),Omi(26),Oma(26)
C
       Character file1*15, cmdbuf*256,file2*15
C
       type*,'min and max Id as 2i6'
       accept 115, id1,id2
C
c
       type*, 'Name of output file'
       accept 110, file1
  110  format(a15)
C
  115  format(2i6)
C
       open(unit=21, file=file1,status='new')
C
       mt=1
       ms=2
       mo=3
       ncount=0
       call fdberrhandle(err_handler)
       call fdbmsghandle(msg_handler)
       login=fdblogin()
       call fdbsetluser(login,'SOCEAN')
       call fdbsetlpwd(login,'Victor')
       dbproc=fdbopen(login,NULL)
       call fdbuse(dbproc,'SouthernOceanDB')
C
       call fdbfcmd(dbproc,'Execute Sel01 %d,%d',Id1,Id2)
       call fdbsqlexec(dbproc)
       call fdbresults(dbproc)
       call fdbbind(dbproc,1,intbind,0,idsel)
       J=0
       do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
       J=J+1
       IDALL(J)=idsel
       end do
C------------------------------------------------------------
       do1 is =1, J
       idsel=idall(is)
       call fdbfcmd(dbproc,'Execute Sel02 %d',idsel)
       call fdbsqlexec(dbproc)
       call fdbresults(dbproc)
       call fdbbind(dbproc,1, intbind,0,depth)
       call fdbbind(dbproc,2,flt8bind,0,T8)
       call fdbbind(dbproc,3,flt8bind,0,S8)
       call fdbbind(dbproc,4,flt8bind,0,O8)
       K=0
       do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
       K=K+1
       z(k)=depth
```

```
        t(k)=sngl(T8)
        s(k)=sngl(S8)
CC        Ox(k)=sngl(O8)
        end do
        nz=k
        do 2 k=1,nz
        if(z(k).gt.400) go to 2
        if(z(k).ge.200. and. z(k).le.400. and.s(k).lt.33.)goto23
        if(z(k).lt.200.and.z(nz).gt.200.and.s(k).lt.29.)go to 23
        go to 2
   23 ncount=ncount+1
        write(21,100) IDALL(is), ms,z(k),s(k)
        type*,ncount,IDALL(is)
    2  continue
C
  107 format(2x,i5)
  100 format(2x,i6,1x,i1,1x,i4,1x,f8.3)
C-----------------------------------------------
    1 continue
C
      close(21)
      stop ' E N D '
      end
C ------------------------------------------

C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
C      EXTERNAL        err_handler
C      EXTERNAL        msg_handler
C
      INTEGER*4       dbproc
      INTEGER*4       severity
      INTEGER*4       errno
      INTEGER*4       oserrno
      INTEGER*4       length
      INTEGER*4       return_code
C
      CHARACTER*(80)  message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C       Message Handler
C       ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
       INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
      2              msgstate,severity, msgtext)
C
       include '(fsybdb)'
C
       INTEGER*4      dbproc
       INTEGER*4      msgno
       INTEGER*4      msgstate
       INTEGER*4      severity
C
       CHARACTER*80   msgtext
         IF (MSGNO.NE.5701) THEN
C
          type *, 'DataServer message ', msgno,
     2        '   state ', msgstate, '    severity ',
     3        severity,' ', msgtext
C
         END IF
          msg_handler = DBNOSAVE
C
       END
```

Compar 21A

```
      Program compar21A
C      V.Guretsky, AWI, 1990
C      Compares headers of stations for the same ship
C     within Aari and Gordon subsets and writes IDs of
C     possible duplicates
C-----------------------------------------------------------
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
      Integer*4  dbproc, login,return_code,error,NID,
     *IDAAR(5000),IDGOR(5000),ID,Crunug,
     *bdgor4,bdaar4,mobgor4,mobaar4,yegor4,yeaar4,mogor4,moaar4,
     *daaar4,dagor4,  IAA(100)
C
      real*8 logor8,loaar8,lagor8,laaar8
C
      Character file1*15, cmdbuf*256,file2*15
      character*5 Ship1
      character*25 Ship2
      character*29 Shipdb
C
      type*,'Name of the input file'
      accept 101, file1
      open(unit=20, file=file1,status='old')
      type*, 'Name of output file'
      accept 101, file2
      open(unit=21, file=file2,status='new')
C
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C
  100 format(2x,i3,2x,a5,2x,a25,2x,i7)
      read(20,400) nseq, Ship2,Crunug,I,J
  400 format(2x,i4,2x,a25,2x,3i7)
      read(20,111)I
      read(20,111) (IDGOR(k),k=1,I)
      read(20,111)J
      read(20,111) (IDAAR(k),k=1,J)
  111 format(2x,10i7)
C-----------------------------------
  101 format(a15)
      kount=0
      do 4 k=1,I
      do5  L=1,J
c
      M=0
      call fdbfcmd(dbproc,'Execute Comp2 %d,%d',IDGOR(k),IDAAR(L))
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,logor8)
      call fdbbind(dbproc,2,flt8bind,0,loaar8)
      call fdbbind(dbproc,3,flt8bind,0,lagor8)
      call fdbbind(dbproc,4,flt8bind,0,laaar8)
      call fdbbind(dbproc,5,intbind,0,bdgor4)
      call fdbbind(dbproc,6,intbind,0,bdaar4)
      call fdbbind(dbproc,7,intbind,0,mobgor4)
      call fdbbind(dbproc,8,intbind,0,mobaar4)
      call fdbbind(dbproc,9,intbind,0,yegor4)
      call fdbbind(dbproc,10,intbind,0,yeaar4)
      call fdbbind(dbproc,11,intbind,0,mogor4)
      call fdbbind(dbproc,12,intbind,0,moaar4)
```

Compar 21A
/90
Compar

Com_5

```
       call fdbbind(dbproc,13,intbind,0,dagor4)
       call fdbbind(dbproc,14,intbind,0,daaar4)
       call fdbnextrow(dbproc)
C
       if(yeaar4.eq.yegor4)M=M+1
       if(moaar4.eq.mogor4)M=M+1
       if(daaar4.eq.dagor4)M=M+1
       if(bdaar4.eq.bdgor4)M=M+1
       if(mobgor4.eq.mobaar4)M=M+1
       e1=abs(logor8-loaar8)
       e2=abs(lagor8-laaar8)
       if(e1.lt.0.05)M=M+1
       if(e2.lt.0.05)M=M+1
       if(M.GE.5)go to 6
       go to 5
     6 continue
       kount=kount+1
   500 format(2x,10i7)
       write(21,500) kount,IDGOR(k),IDAAR(L)
     5 continue
     4 continue
       close(21)
       close(20)
       call fdbexit()
       stop ' E N D '
       end
C ------------------------------------------
C      Error und Message Handler fuer
C      embedded SQL-Programme. In diesen mit
C      INCLUDE '(ERRMSG)' includen.
C
C      Error Handler
C      -------------
C      ERR_HANDLER - This funtion may be coded within the same program
C      or as a separate file that is compiled/linked.
C
       INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
       include '(fsybdb)'
C
C       EXTERNAL        err_handler
C       EXTERNAL        msg_handler
C
       INTEGER*4       dbproc
       INTEGER*4       severity
       INTEGER*4       errno
       INTEGER*4       oserrno
       INTEGER*4       length
       INTEGER*4       return_code
C
       CHARACTER*(80)  message
C
          length = fdberrstr(errno,message)
          type *, 'DB-LIBRARY error: ', message
C
C      Check for operating system errors
C
          length = 0
          message = ' '
          length = fdboserrstr(oserrno, message)
C
          if (oserrno .ne. DBNOERR) then
             type *, 'Operating-system error: ', message
          end if
C
          return_code = fdbdead(dbproc)
```

```
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C       Message Handler
C       ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2             msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4      dbproc
      INTEGER*4      msgno
      INTEGER*4      msgstate
      INTEGER*4      severity
C
      CHARACTER*80   msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
```

Compardel1

```
      Program compardel1
C     This program reads  Id of stations to be deleted for the
C      Gordon Cruise, specified by  Cruise_Number
C          and then deletes them making protocol and infofile
C      V.Guretsky, AWI, May, 1990
C      INPUT FILE - result of program COMPAR3
C-------------------------------------------------------------
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
C
      Integer*4  ncount, dbproc, login,return_code,error,nc,
     *IDgor, IDaar, Crunug,NPERCENT,CRSPEC
C
      Character file1*15, cmdbuf*256, ship*25, file2*15, Date*20,
     *file3*15
C               -------I N P U T------
           type*, 'Name of intput file of station IDs todelete'
      accept 100, file1
  100 format(a15)
      open(unit=20, file=file1,status='old')
C
           type*,'Name of output file for the protocol of deleation'
      accept 100, file2
      open(unit=21,file=file2,status='new')
C
           Type*,'Name of info-file'
      accept 100, file3
      open(unit=23,file=file3,status='new')
C
           Type*,'Insert Date_Time  of transaction as Character*20'
      accept 101, Date
  101 format(A20)
           type*,'Insert Gordon Cruise_Number  (I4)'
      accept 102,Crspec
  102 format(i5)
C     ----------------------------------------
      NNNN=0
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C
C             WRITE HEAD OF THE PROTOCOL
      write(21,201)
      write(21,202) Date
      write(21,204)
  202 format(15x, a20)
  201 format(2x,' PROTOCOL OF DELETION OF STATIONS WITHIN SoOceanDB')
  204 format(2x,'Prog. Compardel1: deleted are stat-s having
     * > 60% identical levels for T, S and Oxg')
C
C            WRITE HEADER OF THE INFOR-FILE
      write(23,205)
      write(23,202) Date
      write(23,204)
  205 format(2x,' Information on the deletion of stations')
C
      kount1=0
  113 continue
      read(20,400, end=112) nseq,Ship,Crunug,NGOR,NAAR
      type 400,nseq,Ship,Crunug,NGOR,NAAR
C
```

Com-4

```
      if(Crunug.ne.Crspec) go to 134
      NNNN=1
  400 format(2x,i4,2x,a25,2x,3i7)
      write(21,400)nseq,Ship,Crunug,NGOR,NAAR
      write(23,400)nseq,Ship,Crunug,NGOR,NAAR
C
C
  134 continue
      kount2=0
  114 continue
      read(20,600,err=115,end=115)jdel, idgor, IDAAR
  600 format(2x,10i7)
      if(IDAAR.gt.40000) go to 114
C
C ********************************
      if(Crunug.ne.Crspec) go to 114
C++++++++++++++++++++++++++++++++++++++++++
      call fdbfcmd(dbproc,'Execute Deletel %d', IdAAR)
      call fdbsqlexec(dbproc)
C+++++++++++++++++++++++++++++++++++++++++
      kount2=kount2+1
      kount1=kount1+1
      write(21,200) kount2,IDaar, Ship
      type 200,kount2
  200 format(2X, 2i7, 2x, a25)
      go to 114
  115 continue
      if(Crunug.ne.Crspec) go to 125
      write(23,301) kount2
      NPERCENT=100*kount2/NGOR
      write(23,306)NPERCENT
  306 format(48x,'This corresponds to',i7,' percents of Gordon
     * stations for this Cruise)')
  301 format(49x,i7,' Aari stations deleted')
      if(NNNN.eq.1)go to 112
  125 continue
      backspace(20)
      go to 113
  112 continue
  302 format(10x/10x/10x,i7,' is total number of deleted stations')
      close(21)
      close(20)
      close(23)
      call fdbexit()
      stop ' E N D '
      end
C -----------------------------------------
C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
C      EXTERNAL        err_handler
C      EXTERNAL        msg_handler
C
      INTEGER*4        dbproc
      INTEGER*4        severity
      INTEGER*4        errno
      INTEGER*4        oserrno
      INTEGER*4         length
      INTEGER*4         return_code
C
      CHARACTER*(80)   message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
C      ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2            msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4        dbproc
      INTEGER*4        msgno
      INTEGER*4        msgstate
      INTEGER*4        severity
C
      CHARACTER*80     msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '    state ', msgstate, '     severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
```

Compar 22

```
      Program compar22
C      V.Guretsky, AWI, 1991,  3 April
C      Compares headers of stations for the same ship
C     within Aari and Gordon subsets and writes IDs of
C     possible duplicates
C
C Comparison is made only for dates!!!!!!!!!!!!!!!!!!!!
C-------------------------------------------------------------
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
      Integer*4  dbproc, login,return_code,error,NID,
     *IDAAR(5000),IDGOR(5000),ID,Crunug,
     *bdgor4,bdaar4,mobgor4,mobaar4,yegor4,yeaar4,mogor4,moaar4,
     *daaar4,dagor4,  IAA(100)
C
      real*8 logor8,loaar8,lagor8,laaar8
C
      Character file1*15, cmdbuf*256,file2*15
      character*5 Ship1
      character*25 Ship2
      character*29 Shipdb
C
      open(unit=20, file='C4.DAT',status='old')
      open(unit=21, file='COM22.DAT',status='new')
C
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C
  200 continue
  100 format(2x,i3,2x,a5,2x,a25,2x,i7)
      read(20,400,end=333) nseq, Ship2,Crunug,I,J
  400 format(2x,i4,2x,a25,2x,3i7)
      read(20,111)J
      read(20,111) (IDAAR(k),k=1,J)
      read(20,111)I
      read(20,111) (IDGOR(k),k=1,I)
  111 format(2x,10i7)
  101 format(a15)
CWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
      write(21,400) nseq,Ship2,Crunug,I,J
CWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
      kount1=0
           do 4 k=1,I
      kount=0
           do5  L=1,J
C
      call fdbfcmd(dbproc,'Execute CAAR %d',IDAAR(L))
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,NID)
      call fdbnextrow(dbproc)
      if(NID.lt.1)GO TO 5   ! if no data for this ID go to next station
C
      M=0
      call fdbfcmd(dbproc,'Execute Comp2 %d,%d',IDGOR(k),IDAAR(L))
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,logor8)
      call fdbbind(dbproc,2,flt8bind,0,loaar8)
C
```

Com3

```
        call fdbbind(dbproc,3,flt8bind,0,lagor8)
        call fdbbind(dbproc,4,flt8bind,0,laaar8)
C
        call fdbbind(dbproc,5,intbind,0,bdgor4)
        call fdbbind(dbproc,6,intbind,0,bdaar4)
C3
        call fdbbind(dbproc,7,intbind,0,mobgor4)
        call fdbbind(dbproc,8,intbind,0,mobaar4)
C4
        call fdbbind(dbproc,9,intbind,0,yegor4)
        call fdbbind(dbproc,10,intbind,0,yeaar4)
C5
        call fdbbind(dbproc,11,intbind,0,mogor4)
        call fdbbind(dbproc,12,intbind,0,moaar4)
C6
        call fdbbind(dbproc,13,intbind,0,dagor4)
        call fdbbind(dbproc,14,intbind,0,daaar4)
C7
        call fdbnextrow(dbproc)
C
        if(yeaar4.eq.yegor4)M=M+1
        if(moaar4.eq.mogor4)M=M+1
        if(daaar4.eq.dagor4)M=M+1
CCC        if(bdaar4.eq.bdgor4)M=M+1
CCC        if(mobgor4.eq.mobaar4)M=M+1
CCC        e1=abs(logor8-loaar8)
CCC        e2=abs(lagor8-laaar8)
CCC        if(e1.le.0.02)M=M+1
CCC        if(e2.le.0.02)M=M+1
CCC        if(M.GE.5)go to 6
        if(M.EQ.3)go to 6
        go to 5
    6 continue
        kount=kount+1
        IAA(kount)=IDAAR(L)
    5 continue
C
        if(kount.eq.0)go to 4
             do m=1,kount
             kount1=kount1+1
CWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
             write(21,500)kount1,IDGOR(k),IAA(m)
CWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
CCC               type 500,kount1,IDGOR(k),IAA(m)
             end do
  500 format(2x,10i7)
    4 continue
      go to 200
  333 continue
      close(21)
      close(20)
      call fdbexit()
      stop ' E N D '
      end
C ------------------------------------------
C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
```

```
      include '(fsybdb)'
C
C      EXTERNAL         err_handler
C      EXTERNAL         msg_handler
C
      INTEGER*4        dbproc
      INTEGER*4        severity
      INTEGER*4        errno
      INTEGER*4        oserrno
      INTEGER*4        length
      INTEGER*4        return_code
C
      CHARACTER*(80)   message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
C      ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2             msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4      dbproc
      INTEGER*4      msgno
      INTEGER*4      msgstate
      INTEGER*4      severity
C
      CHARACTER*80   msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3       severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
```

Compar 3

```
      Program compar3
C      V.Guretsky, AWI, 1990
C  Analogue of compar3 but only for T OR S (NO OXYGEN CONSIDERED!)
C-----------------------------------------------------------
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
      Integer*4  dbproc, login,return_code,error,NID,
     *IDAAR(5000),IDGOR(5000),ID,Crunug, Z4,
     *bdgor4,bdaar4,mobgor4,mobaar4,yegor4,yeaar4,mogor4,moaar4,
     *daaar4,dagor4,  IAA(100)
C
      real*8 logor8,loaar8,lagor8,laaar8,z8,T8A,S8A,O8A,O8G,T8G,S8G
      real*4 z(42),tg1(42),sg1(42),og1(42),ta1(42),sa1(42),oa1(42)
C
      Character file1*15, cmdbuf*256,file2*15,file3*15
      character*5 Ship1
      character*25 Ship2
      character*29 Shipdb
C
      IB=0
C
      type*,'Name of the input file'
      accept 101, file1
      open(unit=21, file=file1,status='old')
      type*, 'Name of the first outputfile'
C
      accept 101, file2
      open(unit=22, file=file2,status='new')
  101 format(a15)
C
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
      LLL=0
C
  200 continue
      read(21,400,end=333) nseq,Ship2,Crunug,I,J
      write(22,400)nseq,Ship2,Crunug,I,J
  400 format(2x,i4,2x,a25,2x,3i7)
      type400,nseq,Ship2, Crunug,I,J
C
      read(21,411)J
      read(21,411)(IDAAR(k),k=1,J)
      read(21,411)I
      read(21,411)(IDGOR(k),k=1,I)
  411 format(2x,10i7)
C        LOOP WITHIN CRUISE FOR GORDON STATIONS
      KOUNT1=0
C
      do2 LGOR=1,I
CCCC        type*,'Lgor=',lgor,'    Ship=',Ship2
  600 format(2x,10i7)
C
      do 2 LAAR=1,J
C
C FIND WETHER AARI STATION IS PRESENT
      call fdbfcmd(dbproc,'Execute Countid %d', IDAAR(LAAR))
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,ID)
      call fdbnextrow(dbproc)
```

Com 2

```
      if(ID.EQ.0)GO TO 2
C
C NOW SELECT DATA FOR BOTH STATIONS
      call fdbfcmd(dbproc,'Execute Stadacom %d,%d',
     * IDGOR(LGOR),IDAAR(LAAR))
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,Z4)
      call fdbbind(dbproc,2,flt8bind,0,T8G)
      call fdbbind(dbproc,3,flt8bind,0,S8G)
      call fdbbind(dbproc,4,flt8bind,0,O8G)
      call fdbbind(dbproc,5,flt8bind,0,T8A)
      call fdbbind(dbproc,6,flt8bind,0,S8A)
      call fdbbind(dbproc,7,flt8bind,0,O8A)
      L=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      L=L+1
      tg1(L)=sngl(T8G)
      sg1(L)=sngl(S8G)
      Og1(L)=sngl(O8G)
      ta1(L)=sngl(T8A)
      sA1(L)=sngl(S8A)
      OA1(L)=sngl(O8A)
      end do
C
C     MAKE COMPARISON OF DATA
      if(L.lt.1)go to 2
      mtp=0
      msp=0
      mt=0
      ms=0
      NT=0
      NS=0
      do 221 KK=1,L
      if(tg1(kk).lt.-2.2.or.tg1(kk).gt.30.) go to 221
      if(ta1(kk).lt.-2.2.or.ta1(kk).gt.30.) go to 221
      NT=NT+1
      if(abs(tg1(Kk)-ta1(kk)).le.0.01) mt=mt+1
  221 continue
      do 222 KK=1,L
      if(sg1(kk).lt.30.0.or.sg1(kk).gt.36.) go to 222
      if(sa1(kk).lt.30.0.or.sa1(kk).gt.36.) go to 222
      NS=NS+1
      if(abs(sg1(kk)-sa1(kk)).le.0.005) ms=ms+1
  222 continue
      NTT=NT*100/L !check made only when dummy values <50% of total
      if(NTT.lt.50) go to 225
      mtp=mt*100/NT
  225 continue
      NSS=NS*100/L ! check made only when dummy values <50% of total
      if(NSS.LT.50)go to 226
      msp=ms*100/NS
  226 continue
      if(mtp.gt.70.or.msp.gt.70)go to444
      go to 2
  444 continue
      kount1=kount1+1
      write(22,600)KOUNT1,IDGOR(lgor),IDAAR(LAAR)
      type 600, kount1,idgor(lgor),IDAAR(LAAR)
    2 continue
C
      go to 200
  333 continue
      close(21)
      close(22)
      call fdbexit()
      stop ' E N D '
      end
C ------------------------------------------
C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
C      EXTERNAL         err_handler
C      EXTERNAL         msg_handler
C
      INTEGER*4        dbproc
      INTEGER*4        severity
      INTEGER*4        errno
      INTEGER*4        oserrno
      INTEGER*4        length
      INTEGER*4        return_code
C
      CHARACTER*(80)   message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
C      ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2           msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4        dbproc
      INTEGER*4        msgno
      INTEGER*4        msgstate
      INTEGER*4       severity
C
      CHARACTER*80    msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2        '   state ', msgstate, '     severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
```

Compar-32

```
       Program compar32
C       V.Guretsky, AWI, 1990
C  Analogue of compar3 but only for T and S (NO OXYGEN CONSIDERED!)
C-------------------------------------------------------
       EXTERNAL err_handler
       External msg_handler
       include '(fsybdb) '
       Integer*4  dbproc, login,return_code,error,NID,
      *IDAAR(5000),IDGOR(5000),ID,Crunug,
      *bdgor4,bdaar4,mobgor4,mobaar4,yegor4,yeaar4,mogor4,moaar4,
      *daaar4,dagor4,  IAA(100)
C
       real*8 logor8,loaar8,lagor8,laaar8,z8,T8A,S8A,O8A,O8G,T8G,S8G
       real*4 z(42),tg1(42),sg1(42),og1(42),ta1(42),sa1(42),oa1(42)
C
       Character file1*15, cmdbuf*256,file2*15,file3*15
       character*5 Ship1
       character*25 Ship2
       character*29 Shipdb
C
       IB=0
C
       type*,'Name of the input file'
       accept 101, file1
       open(unit=21, file=file1,status='old')
       type*, 'Name of the first outputfile'
C
       accept 101, file2
       open(unit=22, file=file2,status='new')
   101 format(a15)
C
       call fdberrhandle(err_handler)
       call fdbmsghandle(msg_handler)
       login=fdblogin()
       call fdbsetluser(login,'SOCEAN')
       call fdbsetlpwd(login,'Victor')
       dbproc=fdbopen(login,NULL)
       call fdbuse(dbproc,'SouthernOceanDB')
       LLL=0
C
   200 continue
       read(21,400,end=333) nseq,Ship2,Crunug,I,J
       write(22,400)nseq,Ship2,Crunug,I,J
   400 format(2x,i4,2x,a25,2x,3i7)
       type400,nseq,Ship2, Crunug,I,J
C
       read(21,411)J
       read(21,411)(IDAAR(k),k=1,J)
       read(21,411)I
       read(21,411)(IDGOR(k),k=1,I)
   411 format(2x,10i7)
C         LOOP WITHIN CRUISE FOR GORDON STATIONS
       KOUNT1=0
C
       do2 LGOR=1,I
C
C
C

       if(IDGOR(LGOR).EQ.100453)IB=99
       if(IB.EQ.0)go to 2
C
C
C
CCCC        type*,'Lgor=',lgor,'   Ship=',Ship2
   600 format(2x,10i7)
```

Com-1

```
C
       do 2 LAAR=1,J
       call fdbfcmd(dbproc,'Execute Stadacom %d,%d',
      * IDGOR(LGOR),IDAAR(LAAR))
       call fdbsqlexec(dbproc)
       call fdbresults(dbproc)
       call fdbbind(dbproc,1,flt8bind,0,Z8)
       call fdbbind(dbproc,2,flt8bind,0,T8G)
       call fdbbind(dbproc,3,flt8bind,0,S8G)
       call fdbbind(dbproc,4,flt8bind,0,O8G)
       call fdbbind(dbproc,5,flt8bind,0,T8A)
       call fdbbind(dbproc,6,flt8bind,0,S8A)
       call fdbbind(dbproc,7,flt8bind,0,O8A)
       L=0
       do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
       L=L+1
       z(L)=sngl(Z8)
       tg1(L)=sngl(T8G)
       sg1(L)=sngl(S8G)
       Og1(L)=sngl(O8G)
       ta1(L)=sngl(T8A)
       sA1(L)=sngl(S8A)
       OA1(L)=sngl(O8A)
       end do
C
C      COMPARISON OF DATA
       if(L.lt.1)go to 2
       mt=0
       ms=0
       do 22 KK=1,L
       if(abs(tg1(Kk)-ta1(kk)).lt.0.005) mt=mt+1
       if(abs(sg1(kk)-sa1(kk)).lt.0.005) ms=ms+1
    22 continue
       mtp=mt*100/L
       msp=ms*100/L
       if(mtp.gt.60.and.msp.gt.60)go to444
       go to 2
   444 continue
       kount1=kount1+1
       write(22,600)KOUNT1,IDGOR(lgor),IDAAR(LAAR)
       type 600, kount1,idgor(lgor),IDAAR(LAAR)
     2 continue
C
       go to 200
   333 continue
       close(21)
       close(22)
       call fdbexit()
       stop ' E N D '
       end
C -----------------------------------------
C      Error und Message Handler fuer
C      embedded SQL-Programme. In diesen mit
C      INCLUDE '(ERRMSG)' includen.
C
C      Error Handler
C      -------------
C      ERR_HANDLER - This funtion may be coded within the same program
C      or as a separate file that is compiled/linked.
C
       INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
       include '(fsybdb)'
C
C       EXTERNAL        err_handler
C       EXTERNAL        msg_handler
```

```
C
       INTEGER*4          dbproc
       INTEGER*4          severity
       INTEGER*4          errno
       INTEGER*4          oserrno
       INTEGER*4          length
       INTEGER*4          return_code
C
       CHARACTER*(80)     message
C
          length = fdberrstr(errno,message)
          type *, 'DB-LIBRARY error: ', message
C
C      Check for operating system errors
C
          length = 0
          message = ' '
          length = fdboserrstr(oserrno, message)
C
          if (oserrno .ne. DBNOERR) then
             type *, 'Operating-system error: ', message
          end if
C
          return_code = fdbdead(dbproc)
C
          if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2       (severity .eq. EXSERVER)) then
             err_handler = INT_EXIT
C
          else
             err_handler = INT_CANCEL
          end if
C
          END
C
C       Message Handler
C       ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
       INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2             msgstate,severity, msgtext)
C
       include '(fsybdb)'
C
       INTEGER*4        dbproc
       INTEGER*4        msgno
       INTEGER*4        msgstate
       INTEGER*4        severity
C
       CHARACTER*80     msgtext
         IF (MSGNO.NE.5701) THEN
C
          type *, 'DataServer message ', msgno,
     2        '   state ', msgstate, '    severity ',
     3        severity,' ', msgtext
C
         END IF
          msg_handler = DBNOSAVE
C
       END
```

```
      PROGRAM DUPLIC23
C     SELECTION OF PAIRS OF DUPLICATE STATIONS AND WRITING THEM
C     INTO COMBINED TABLE
C     V.GURETSKY, APREL 1990, A W I
C
      EXTERNAL                  err_handler
      EXTERNAL                  msg_handler
C
       include '(fsybdb)'
C
C     Variablendeklaration
C     --------------------
      Real z(50,2),t(50,2),s(50,2),O2(50,2),modepth(2),depth(2),
     *lon(2), lat(2)
C
      REAL*8 lon8,lat8,depth8,modepth8,z8,t8,s8,ox8
C
      CHARACTER cmdbuf*256, finp*15, fout*15, ship1*15, ship2*15
C
      INTEGER*2 nz(2),numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2),nms(2), numer
C
      INTEGER*4
     * login
     * ,dbproc
     * ,return_code
     * ,error
     *, stnum,year, month, day, hour, numobs, msq, id, nc,
     * id1,id2,nc1,nc2,nc3,nc4,nstc1, nstc2
C     -------------------------------------------
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
C     ----------------------------------------------
      login = fdblogin()
      call fdbsetluser(login, 'SOCEAN')
      call fdbsetlpwd(login, 'Victor')
      dbproc = fdbopen(login, NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
C     -------------------------------------
C       I N P U T
      TYPE*, 'NAME OF INPUT FILE '
      read(6,100) finp
  100 format(a15)
      open(unit=20, file=finp, status='old')
        type*, 'Name of output file '
        read(6,100)fout
        open(unit=21, file=fout,status='new')
  200 continue
C     --------------------------------
      read(20,111,end=112) nnnn,id1,id2,nc1,nc2,Ship1,Ship2,
     *nstc1,nstc2
      TYPE*,nnnn,id1,id2,nstc1,nstc2
      if(nnnn.gt.2) go to 112
C    ------------------------------------
      nc3=nc1
      nc4=nc2
      if(nc1.lt.0)nc3=30000-nc1
      if(nc2.lt.0)nc4=30000-nc2
C     --------------------
C      CALLS OF THE STORED PROCEDURE  Ship
      call fdbfcmd(dbproc,'Execute Ship %d', nc3)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,charbind,0,ship1)
      call fdbnextrow(dbproc)
         call fdbfcmd(dbproc,' Execute Ship %d',nc4)
```

Duplicates

Dup-15

```
          call fdbsqlexec(dbproc)
          call fdbresults(dbproc)
          call fdbbind(dbproc,1,charbind,0,ship2)
          call fdbnextrow(dbproc)
C ------------------------
      Type*, 'num=',nnnn
  111 format(2x,5i7,2x,a15,2x,a15,2x,2i7)
      id=id1
      do300 ii=1,2
      if(ii.eq.2)id=id2
C
      call fdbfcmd(dbproc,'Execute Dup34 %d', id)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,stnum)
      call fdbbind(dbproc,2,intbind,0,year)
      call fdbbind(dbproc,3,intbind,0,month)
      call fdbbind(dbproc,4,intbind,0,day)
      call fdbbind(dbproc,5,intbind,0,hour)
      call fdbbind(dbproc,6,flt8bind,0,lon8)
      call fdbbind(dbproc,7,flt8bind,0,lat8)
      call fdbbind(dbproc,8,flt8bind,0,depth8)
      call fdbbind(dbproc,9,flt8bind,0,Modepth8)
      call fdbbind(dbproc,10,intbind,0,numobs)
      call fdbbind(dbproc,11,intbind,0,msq)
      call fdbnextrow(dbproc)
C          Umwandlung von REAL*8 Variablen auf REAL
C          ---------------------------------------
           LON(ii)     = sngl(LON8)
           LAT(ii)     = sngl(LAT8)
           DEPTH(ii) = sngl(DEPTH8)
           MODEPTH(ii) = sngl(MODEPTH8)
           numst(ii)=stnum
           nyear(ii)=year
           nmonth(ii)=month
           nday(ii)=day
           nhour(ii)=hour
           nob(ii)=numobs
           nms(ii)=msq
C    --------------------------------------------------
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,n)
      call fdbnextrow(dbproc)
      nz(ii)=n
C     -----------------------------------------------------
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,z8)
      call fdbbind(dbproc,2,flt8bind,0,t8)
      call fdbbind(dbproc,3,flt8bind,0,s8)
      call fdbbind(dbproc,4,flt8bind,0,ox8)
          j = 0
          do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
          J=J+1
          z(j,ii)=sngl(z8)
          t(j,ii)=sngl(t8)
          s(j,ii)=sngl(s8)
          O2(j,ii)=sngl(ox8)
          end do
  300 continue
C         REARRANGMENT OF TABLES
      N2=NZ(2)
      n1=nz(1)
      if(z(1,2)-z(1,1)) 70,76,71
   70 continue
C      Upper level of the first station is deeper
      m=0
      do 72 k=1,n2
      m=m+1
   72 if(z(1,1).eq.z(k,2)) go to 73
   73 k1=n1+m
      k2=n1+1
      do 74 k=1,n1
      z(k1-k,1) = z(k2-k,1)
      t(k1-k,1) = t(k2-k,1)
      s(k1-1,1)=  s(k2-k,1)
   74 O2(k1-k,1)=O2(k2-k,1)
      do 75 k=1,m
      z(k,1)=99.
      t(k,1)=99.
      s(k,1)=99.
   75 O2(k,1)=99.
      go to 76
   71 continue
C     Upper level of the second station is deeper
      m=0
      do 92 k=1,n1
      m=m+1
   92 if(z(1,2).eq.z(k,1)) go to 93
   93 k1=n2+m
      k2=n2+1
      do 94 k=1,n2
      z(k1-k,2)=z(k2-k,2)
      t(k1-k,2)=t(k2-k,2)
      s(k1-k,2)=s(k2-k,2)
   94 O2(k1-k,2)=O2(k2-k,2)
      do 95 k=1,m
      z(k,2)=99.
      t(k,2)=99.
      s(k,2)=99.
   95 O2(k,2)=99.
   76 continue
      n=imax0(nz(1),nz(2))
C     ********************************
C                 O U T P U T
      write(21,111)nnnn,nstc1,nstc2
      write(21,111)id1,id2
      write(21,50) nc1,ship1,nc2,ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      nn=numst(2)-numst(1)
      write(21,111) numst,nn
      dd=lon(2)-lon(1)
      write(21,51) Lon, dd
      dd=Lat(2)-Lat(1)
      write(21,51) Lat, dd
   51 format(2x,3f8.3)
      write(21,52) Depth
   52 format(2x,2f7.0)
      write(21,52) Modepth
      write(21,111) nyear
      write(21,111) nmonth
      write(21,111) nday
      nn=nhour(2)-nhour(1)
      write(21,111) nhour,nn
      nn=nob(2)-nob(1)
      write(21,111) nob,nn
      nn=nms(2)-nms(1)
      write(21,111) nms,nn
      write(21,111) n
      do 27 k=1,n
      tt=t(k,2)-t(k,1)
      ss=s(k,2)-s(k,1)
      xx=O2(k,2)-O2(k,1)
   27 write(21,55) z(k,1),(t(k,j),j=1,2),tt,(s(k,j),j=1,2),ss,
     *(O2(k,j),j=1,2),xx,z(k,2)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2,1x,f5.0)
      go to 200
C     **********************************
  112 continue
      close(unit=21)
      call fdbexit()                       ! Schliessen der DB-Library
C
      CLOSE(UNIT=20)
      stop '********* E N D *********'
      END
C
C -----------------------------------------

C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
C      EXTERNAL         err_handler
C      EXTERNAL         msg_handler
C
      INTEGER*4        dbproc
      INTEGER*4        severity
      INTEGER*4        errno
      INTEGER*4        oserrno
      INTEGER*4        length
      INTEGER*4        return_code
C
      CHARACTER*(80)   message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
```

```
C       ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C              or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2           msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4      dbproc
      INTEGER*4      msgno
      INTEGER*4      msgstate
      INTEGER*4      severity
C
      CHARACTER*80   msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
```

```
                 Program duplic13
C     Correction of duplicate stations
C     This program makes correction of duplicates,
C        Plus-minus 0.1 degree coordinate criterium
C       ( as was used in the program Duplic9)
C      Correction of T, S, Ox is made through the terminal
C     Checking of stability is possible also
C     This is a slight modification of Duplic12
C            V.Guretsky, May, 1990, AWI
C     ------------------------------------
      real lon(2), lat(2), z(50), s(50,2),O2(50,2),t(50,2),
     *depth(2), modepth(2), dt(50),ds(50),dox(50),
     * sigt(50,2), sigpot(50,2), pbar(50), tpot(50,2), dsig(50,2),
     * dtp(50,2), dtdt(50,2),
     * sr(50),tr(50),Or(50),lonr,latr,modepthr, sig0(50,2)
C
      integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
      character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15, x*1

      integer*4 nc(2), id(2), ncr, idr
C -------------------------------------------------------------------
      ncount=0
C          I N P U T
      type*, 'Name of input file'
      accept 100, file1
  100 format(a12)
      open(unit=21, file=file1,status='old')
      type *, 'Name of deleated stations file'
      accept 100, file2
      type*, 'Name of remained station file'
      accept 100, file3
  555 continue
      read(21,111,end=112) nnn
      read(21,111) id
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
   52 format(2x,2f7.0)
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55)z(k),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2)
      type*,nnn,n
C        CHECK IF THIS PAIR HAS ALREADY BEEN PROCESSED
      if(depth(1).lt.0..and.depth(2).lt.0.) goto 555
  111 format(2x,5i7)
C      ------------------------------------------------------
C        Coordinates   Criterium for duplicates
      if(abs(dlon).ge.0.1) go to 555
    2 if(abs(dlat).ge.0.1) go to 555
C      ------------------------------------------------------
C           T Y P E   S T A T I O N S   O N   T H E   S C R E E N
```

Dup-14

```
  444 continue
      type 111, nnn
      type 111, id
      type 50, nc(1),Ship1, nc(2),ship2
      type 111, numst
      type 51, Lon, dlon
      type 51, Lat, dlat
      type 52, Depth
      type 52, Modepth
      type 111, nyear
      type 111, nmonth
      type 111, nday
      type 111, nhour
      type 111, nob
      type 111, nms
      type 111, n
C     -----------------------------
C      type*,'$$$$$                   Further?- any symbol'
C      accept 56,x
   56 format(a1)
C      ----------------------------
      do 28 k = 1, n
   28 type 55, z(k), (t(k,j),j=1,2), dt(k), (s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k)
C      ----------------------------
      type*,'$$$$$  type station again? 0 - no  1 - yes'
      accept 57,k
      if(k)445,445,444
   57 format(2i1)
  445 continue
      type*,'$$$$$  No correction for this pair - 0; YES  1'
      accept 57, k
      if(k) 555,555,557
  557 continue
C     -----------------------------------------
C     CORRECTION OF DUMMY VALUES FOR LEVEL DEPTH

C     ----------------------------
C        CHECK STABILITY OF WATER COLUMN
C      type*,'$$$$$    Want to check stability? 0 - no   1 - yes '
C      accept 57,k
C      if(k)446,446,447
  447 do 85 k=2,n
      do 85 i=1,2
      pbar(k)=z(k)/10.  ! Pres. in Bars
      tpot(k,i)=pttmpr(s(k-1,i), t(k-1,i), z(k-1),z(k))
      dtdt(k,i)=tpot(k,i)-t(k-1,i)
      sigpot(k,i)=(1./ALPHA(pbar(k),tpot(k,i),s(k-1,i)))-1000.
      sigt(k,i)=(1./ALPHA(pbar(k),t(k,i),s(k,i)))-1000.
      dsig(k,i)=sigpot(k,i)-sigt(k,i)
      dtp(k,i)=tpot(k,i)-t(k,i)
      sig0(k,i)=(1./ALPHA(0.,t(k,i),s(k,i)))-1000.
   85 continue
      n1=0
      n2=0
      nd1=0
      nd2=0
C      type 211
  211 format(2x,'  z  ',' sigt  ','sigpot','   sigpot-sigt')
C------------------------------------------------
C      CALCULATION OF DUMMY AND INVERSION NUMBERS
       do 87k=2,n
C      type 212, z(k),sigt(k,1),sigpot(k,1),dsig(k,1),
C     *sigt(k,2),sigpot(k,2),dsig(k,2)
      if(t(k,1).lt.-2.3.or.t(k,1).gt.30..or.s(k,1).lt.30..or.
     *   s(k,1).gt.36..or.sig0(k,1).lt.25..or.sig0(k,1).gt.30.)
     *    nd1=nd1+1
      if(t(k,2).lt.-2.3.or.t(k,2).gt.30..or.s(k,2).lt.30..or.
     *    s(k,2).gt.36..or.sig0(k,2).lt.25..or.sig0(k,2).gt.30.)
     *    nd2=nd2+1
  212 format(2x,f5.0,1x,3(1x,f7.3),4x,3(1x,f7.3))
      if( abs(dsig(k,1)).gt.0.5)goto 449
      if(dsig(k,1).gt.0.) n1=n1+1
  449 if(abs(dsig(k,2)).gt.0.5) goto 87
      if(dsig(k,2).gt.0.) n2=n2+1
   87 continue
      type 213, n1, n2
  213 format(2x,'Number of inversions: first ',i2,'  second ',i2)
      type 214, nd1, nd2
  214 format(2x,'Number of dummy     : first ',i2,'  second ',i2)
  446 continue
C     ---------------------------
C        W H I C H  S T A T I O N   T O  K E E P AND  D E L E A T E
      type*,'$$$$$ Type  which station  to keep'
      accept 57,jjj
      shipd=ship2
      shipk=ship1
      mmm=2
      if(jjj.eq.2) mmm=1
      if(jjj.eq.2)shipd=ship1
      if(jjj.eq.2)shipk=ship2
    4 continue
C     ----------------------------------
      idr=id(jjj)
      ncr=nc(jjj)
      lonr=(lon(1)+lon(2))/2.
      latr=(lat(1)+lat(2))/2.
      numstr=numst(jjj)
      if(numst(jjj).le.0.or.numst(jjj).gt.500) numstr=numst(mmm)
      depthr=depth(jjj)
      modepthr=modepth(jjj)
      nyearr=nyear(jjj)
      nmonthr=nmonth(jjj)
      ndayr=nday(jjj)
      nhourr=max0(nhour(1),nhour(2))
      nobr=max0(nob(1),nob(2))
C       NOW OBJECTIVE CORRECTION OF OXYGEN TAKES PLACE
      do7 k = 1,n
      Or(k)=99.
      if(O2(k,1).lt.0..or.O2(k,1).gt.15.) go to 701
      go to 702
  701 continue
      Or(k)=O2(k,2)
      goto 7
  702 if(O2(k,2).lt.0..or.O2(k,2).gt.15.) go to 703
      go to 704
  703 Or(k)=O2(k,1)
      go to 7
  704 continue
      r=Abs(O2(k,1)-O2(k,2)) - 70.
      if(r)8,8,9
    8 Or(k)=amax1(O2(k,1),O2(k,2))
      goto 7
    9 Or(k)=amin1(O2(k,1),O2(k,2))
    7 continue
C     ________-----------------------------------------
      type*,' Want to correct numbers of remaning station? 0 - N  1-Y'
      accept 57,k
      if(k)300,300,303
  300 continue
      do 302 k = 1, n
      tr(k)=t(k,jjj)
  302 sr(k)=s(k,jjj)
      go to 301
  303 continue
C       CORRECTION OF TEMPERATURE  for the levels, where difference
C                                  is more than 0.001 degree C
      do 29 k=1,n
      if(abs(dt(k)).le.0.001) go to 29
      type 55, z(k),(t(k,j),j=1,2)
      type*,' $$$$$    which value to accept, 1 or 2 or dummy (3) ?'
      accept 57,ii
      if(ii.eq.3) go to  39
      tr(k)=t(k,ii)
      goto 29
   39 tr(k)=-99.
   29 continue
C      -------------------------------------------------
C        CORRECTION OF S A L I N I T Y for the levels, where difference
c                                     is more than 0.001 permille
      do 41 k=1,n
      if(abs(ds(k)).le.0.001) go to 41
      type 55, z(k),(s(k,j),j=1,2)
      type*,'$$$$$   which value to accept, 1 or 2 or dummy (3) ?'
      accept 57, ii
      if(ii.eq.3) go to 42
      sr(k)=s(k,ii)
      goto41
   42 sr(k)=-99.
   41 continue
      type*, 'Want correct oxygen from the terminal? Yes 1, No- 0'
      accept 57,k
      if(k.le.0) go to 301
C       CORRECTION OF OXYGEN
      do 411 k=1,n
      if(abs(dox(k)).le.0.01) go to 411
      type 55, z(k),(O2(k,ii),ii=1,2)
      type*, '  Which value to accept: 1 or 2 or dummy?'
      accept 57, ii
      if(ii.eq.3) go to 421
      Or(k)=O2(k,ii)
      go to 411
  421 Or(k)=-99.
  411 continue
C   ---------------------------------------------------
  301 continue
      type*,'Are you satisfied? 0 - No  1 - Yes'
      accept 57, k
      if(k.eq.0) go to 444
      ncount=ncount+1
      if(ncount.eq.1)go to 5
      goto6
    5 open(unit=22,file=file2,status='new')
      open(unit=23,file=file3,status='new')
    6 continue
C     -------------------------------------------------
C                          O U T P U T
      write(22,200) ncount,id(mmm),nc(mmm), shipd
  200 format(2x,3i7,2x,a15)
      write(23,111) ncount
      write(23,111)idr
      write(23,50) ncr, shipk
      write(23,111)numstr
      write(23,51)Lonr
      write(23,51)Latr
      write(23,52)Depthr
      write(23,52)Modepthr
      write(23,111)nyearr
      write(23,111) nmonthr
      write(23,111) ndayr
      write(23,111) nhourr
      write(23,111) nobr
      write(23,111)nmsr
      write(23,111) n
      do11 k=1,n
   11 write(23,96) z(k),Tr(k),sr(k),Or(k)
   96 format(2x,f5.0,1x,f8.3,1x,f8.3,1x,f6.2)
      goto555
  112 continue
      close(unit=21)
      close(unit=22)
      close(unit=23)
      stop '*** E N D ***'
      end
      FUNCTION PTTMPR ( SALZ, TEMP, PRES, RFPRES )
C -------------------------------------------------------------------
C Checkwert: PTTMPR = 36.89073 DegC
C       fuer SALZ   =    40.0 psu
C            TEMP   =    40.0 DegC
C            PRES   = 10000.000 dbar
C            RFPRES =     0.000 dbar
C
      PARAMETER ( CT2  = 0.29289322,  CT3  =  1.707106781,
     1            CQ2A = 0.58578644,  CQ2B =  0.121320344,
     2            CQ3A = 3.414213562, CQ3B = -4.121320344 )
C
      P  = PRES
      T  = TEMP
      DP = RFPRES-PRES
      DT = DP*ADLPRT ( SALZ, T, P )
      T  = T + 0.5*DT
      Q  = DT
      P  = P + 0.5*DP
      DT = DP*ADLPRT ( SALZ, T, P )
      T  = T + CT2*(DT-Q)
      Q  = CQ2A*DT + CQ2B*Q
      DT = DP*ADLPRT ( SALZ, T, P )
      T  = T + CT3*(DT-Q)
      Q  = CQ3A*DT + CQ3B*Q
      P  = RFPRES
      DT = DP*ADLPRT ( SALZ, T, P )
      PTTMPR = T + (DT-Q-Q)/6.0
      END
C
C
C -------------------------------------------------------------------
      FUNCTION ADLPRT ( SALZ, TEMP, PRES )
C -------------------------------------------------------------------
C Berechnet aus dem Salzgehalt/psu (SALZ), der in-situ Temperatur/degC
C (TEMP) und dem in-situ Druck/dbar (PRES) den adiabatischen Temperatur-
C gradienten/(K Dbar^-1) ADLPRT.
C Checkwert: ADLPRT =     3.255976E-4 K dbar^-1
C       fuer SALZ   =    40.0 psu
C            TEMP   =    40.0 DegC
C            PRES   = 10000.000 dbar
C
      PARAMETER ( S0 = 35.0,
     1 A0 =  3.5803E-5,  A1 =  8.5258E-6,  A2 = -6.8360E-8,
     2 A3 =  6.6228E-10, B0 =  1.8932E-6,  B1 = -4.2393E-8,
     3 C0 =  1.8741E-8,  C1 = -6.7795E-10, C2 =  8.7330E-12,
     4 C3 = -5.4481E-14, D0 = -1.1351E-10, D1 =  2.7759E-12,
     5 E0 = -4.6206E-13, E1 =  1.8676E-14, E2 = -2.1687E-16 )
C
      DS = SALZ-S0
      ADLPRT = ( ( (E2*TEMP + E1)*TEMP + E0 )*PRES
     1          + ( (D1*TEMP + D0)*DS
     2              + ( (C3*TEMP + C2)*TEMP + C1 )*TEMP + C0 ) )*PRES
     3  + (B1*TEMP + B0)*DS +  ( (A3*TEMP + A2)*TEMP + A1 )*TEMP + A0
      END
C
C
C ---------------------------------------------------------------------
      FUNCTION ALPHA(P,T,S)
C ---------------------------------------------------------------------
C     EQUATION OF STATE FOR SEAWATER PROPOSED BY JPOTS 1980
C       UNITS:
C               PRESSURE        P          BARS
C               TEMPERATURE     T          DEG CELCIUS (IPTS-68)
C               SALINITY        S          NSU (IPSS-78)
C               DENSITY         RHO        KG/M**3
C               SPEC. VOL.      ALPHA      M**3/KG
C     CHECK VALUE:
C               ALPHA = 9.435561E-4 M**3/KG
C               FOR:
C                       S = 40 NSU
C                       T = 40 DEG C
C                       P = 1000 BARS
C PDP11 GETESTET: 0.94355614 E-03
C END OF DOC
      IMPLICIT INTEGER*2 (I-N)
      REAL P,T,S,RHO,SR,R1,R2,R3,R4
      REAL A,B,C,D,E,A1,B1,AW,BW,K,KO,KW
      EQUIVALENCE (E,D,B1,R4),(BW,B,R3),(C,A1,R2)
      EQUIVALENCE (AW,A,R1,RO),(KW,KO,K)
      SR=SQRT(ABS(S))
C PURE WATER DENSITY AT ATM PRESS.
      R1=((((6.536332E-9*T-1.120083E-6)*T+1.001685E-4)*T
     *-9.095290E-3)*T+6.793952E-2)*T+999.842594
C SEAWATER DENSITY AT ATM PRESS.
      R2=(((5.3875E-9*T-8.2467E-7)*T+7.6438E-5)*T-4.0899E-3)*T
     *+8.24493E-1
      R3=(-1.6546E-6*T+1.0227E-4)*T-5.72466E-3
      R4=4.8314E-4
      RHO=(R4*S + R3*SR + R2)*S + R1
C SPECIFIC VOL. AT ATM PRESS
      ALPHA=1.0/RHO
      IF(P.EQ.0.0) RETURN
C COMPUTE SECANT BULK MODULUS K(P,T,S)
      E=(9.1697E-10*T+2.0816E-8)*T-9.9348E-7
      BW=(5.2787E-8*T-6.12293E-6)*T+8.50935E-5
      B=BW + E*S
C
      D=1.91075E-4
      C=(-1.6078E-6*T-1.0981E-5)*T+2.2838E-3
      AW=((-5.77905E-7*T+1.16092E-4)*T+1.43713E-3)*T
     *+3.239908
      A=(D*SR + C)*S + AW
C
      B1=(-5.3009E-4*T+1.6483E-2)*T+7.944E-2
      A1=((-6.1670E-5*T+1.09987E-2)*T-0.603459)*T+54.6746
      KW=(((-5.155288E-5*T+1.360477E-2)*T-2.327105)*T
     *+148.4206)*T+19652.21
C COMPUTE K(0,T,S)
      KO=(B1*SR + A1)*S + KW
C EVALUATE K(P,T,S)
      K=(B*P + A)*P + KO
      ALPHA=ALPHA*(1.0-P/K)
      RETURN
      END
C
```

```
                Program duplic10
C     Correction of duplicate stations
C     This program makes correction of coordinates of duplicates,
C     which are far beyond the limit of plus-minus 0.1 degree,
C     used in the program Duplic9.
C     That is the correction is made only for crude errors in
C     coordinates. Otyher information for the station to be saved
C     is aquired as in the program Duplic9
C            V.Guretsky, May, 1990, AWI
C     ------------------------------------
      real lon(2), lat(2), z(50), s(50,2),O2(50,2),t(50,2),
     *depth(2), modepth(2), dt(50),ds(50),dox(50)
      integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
      character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15
      integer*4 nc(2), id(2)
C     ------------------------------------
      tmin=-2.3
      tmax=30.0
      smin1=27.
      smin2=33.5
      smax=35.2
C---------------------------------
      ncount=0
C          I N P U T
      type*, 'Name of input file'
      accept 100, file1
  100 format(a12)
      open(unit=21, file=file1,status='old')
      type *, 'Name of deleated stations file'
      accept 100, file2
      type*, 'Name of remained station file'
      accept 100, file3
  555 continue
      read(21,111,end=112) nnn
      read(21,111) id
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
   52 format(2x,2f7.0)
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55)z(k),(t(k,j),j=1,2);dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2)
      type*,nnn,n
C       CHECK IF THIS PAIR HAS ALREADY BEEN PROCESSED
      if(depth(1).lt.0..and.depth(2).lt.0.) goto 555
  111 format(2x,5i7)
C     ------------------------------------------------
C        T and S   Criterium for duplicates
      do 2 k = 1,n
      if(abs(dt(k)).ge.0.005) go to 555
    2 if(abs(ds(k)).ge.0.005) go to 555
```

Dup-13

```
C       ------------------------------------------------------
C          Setting of proper coordinates
        type 30, lat(1), lon(1), nms(1), id(1), ship1
        type 31, lat(2), lon(2), nms(2), id(2),ship2
        type 32, dlat, dlon
        type*,'IF NO correction for this pair type 0, if yes type 1'
        accept 33,ij
        if(ij)555,555,556
  556   continue
   30   format(2x,'La1=',f8.3,' Lon1=',f8.3,' msq1=',i3,' ID1=',i7,1x,a15)
   31   format(2x,'La2=',f8.3,' Lon2=',f8.3,' msq2=',i3,' ID2=',i7,1x,a15)
   32   format(6x,f8.3,5x,f8.3)
        type*,'Which Latitude? 1-1, 2-2 '
        accept 33, ij
   33   format(i1)
        if(ij-1) 41,41,42
   42   Lat(1)=Lat(2)
   41   continue
        type*,'Which longitude? 1-1, 2-2 '
        accept 33,ij
        if(ij-1) 61,61,62
   62   lon(1)=lon(2)
   61   continue
C      ----------------------------------------
        jjj=1 ! keep
        mmm=2 ! deleate
        shipk=ship1
        shipd=ship2
        if(abs(numst(2)).lt.Abs(numst(1))) goto3
        goto4
    3   jjj=2 ! keep
        mmm=1 ! deleate
        shipd=ship1
        shipk=ship2
    4   continue
C       save remaining station at the place of the first duplicate
        id(1)=id(jjj)
        nc(1)=nc(jjj)
        ship1=shipk
        numst(1)=numst(jjj)
        depth(1)=depth(jjj)
        modepth(1)=modepth(jjj)
        nyear(1)=nyear(jjj)
        nmonth(1)=nmonth(jjj)
        nday(1)=nday(jjj)
        nhour(1)=max0(nhour(1),nhour(2))
        nob(1)=max0(nob(1),nob(2))
        do7 k = 1,n
        t(k,1)=(t(k,1)+t(k,2))/2.
        s(k,1)=(s(k,1)+s(k,2))/2.
        r=Abs(O2(k,1)-O2(k,2)) - 70.
        if(r)8,8,9
    8   O2(k,1)=amax1(O2(k,1),O2(k,2))
        goto 7
    9   O2(k,1)=amin1(O2(k,1),O2(k,2))
    7   continue
        ncount=ncount+1
        if(ncount.eq.1)go to 5
        goto6
    5   open(unit=22,file=file2,status='new')
        open(unit=23,file=file3,status='new')
    6   continue
C       -------------------------------------------------
C                             O U T P U T
        write(22,200) ncount,id(mmm),nc(mmm), shipd
  200   format(2x,3i7,2x,a15)
```

```
      write(23,111) ncount
      write(23,111)id(1)
      write(23,50) nc(1), shipk
      write(23,111)numst(1)
      write(23,51)Lon(1)
      write(23,51)Lat(1)
      write(23,52)Depth(1)
      write(23,52)Modepth(1)
      write(23,111)nyear(1)
      write(23,111) nmonth(1)
      write(23,111) nday(1)
      write(23,111) nhour(1)
      write(23,111) nob(1)
      write(23,111)nms(1)
      write(23,111) n
      do11 k=1,n
   11 write(23,56) z(k),T(k,1),s(k,1),O2(k,1)
   56 format(2x,f5.0,1x,f8.3,1x,f8.3,1x,f6.2)
      goto555
  112 continue
      close(unit=21)
      close(unit=22)
      close(unit=23)
      stop '*** E N D ***'
      end
      Program duplic83
C     Roll-back for the file
C
C     V.Guretsky, April, 1990, AWI
C     ------------------------------------
      real lon(2), lat(2), z(50), s(50,2),O2(50,2),t(50,2),
     *depth(2), modepth(2), dt(50),ds(50),dox(50)
      integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
      character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15, shipdel*15
      integer*4 nc(2), id(2), ncdel, iddel(800), ndel
C     ------------------------------------
      L=0
C          I N P U T  O F  FILE-NAMES
      type*, 'Name of input file of duplicate stations'
      accept 100, file1
  100 format(a12)
      open(unit=21, file=file1,status='old')
  333 continue
      read(21,111,end=112) nnn
      read(21,111) id
   25 format (2x,3i5)
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
      if(Depth(1).gt.0..and.Depth(2).gt.0.) go to 16
      go to 18
   52 format(2x,2f7.0)
   16 continue
   17 L=L+1
      type*, L
   18 continue
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55)z(k),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2)
  111 format(2x,5i7)
      go to 333
  112 continue
      close(unit=21)
C     ------------------------------------------------------
      stop '*** E N D ***'
      end
```

Dup-12

```
      Program duplic82
C     Flagging of pairs of duplicate stations, which have been processed
C     BOTH depths of duplicate pair change their plus-sign for minus
C     V.Guretsky, April, 1990, AWI
C     ------------------------------------
      real lon(2), lat(2), z(50), s(50,2),O2(50,2),t(50,2),
     *depth(2), modepth(2), dt(50),ds(50),dox(50)
      integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
      character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15, shipdel*15
      integer*4 nc(2), id(2), ncdel, iddel(800), ndel
C     ------------------------------------
C          I N P U T   O F   FILE-NAMES
      L=0
      type*, 'Name of input file of duplicate stations'
      accept 100, file1
  100 format(a12)
      open(unit=21, file=file1,status='old')
  333 continue
      read(21,111,end=112) nnn
      read(21,111) id
   25 format (2x,3i5)
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
      if(Depth(1).le.0..and.Depth(2).le.0.) L=L+1
      type*, L
      read(21,52)Modepth
   52 format(2x,2f7.0)
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55)z(k),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2)
  111 format(2x,5i7)
      go to 333
  112 continue
      close(unit=21)
C     -------------------------------------------------------
      stop '*** E N D ***'
      end
```

Dup-11

```
      Program duplic81
C     Roll-back for the file
C
C     V.Guretsky, April, 1990, AWI
C    ------------------------------------
      real lon(2), lat(2), z(50), s(50,2),O2(50,2),t(50,2),
     *depth(2), modepth(2), dt(50),ds(50),dox(50)
      integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
      character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15, shipdel*15
      integer*4 nc(2), id(2), ncdel, iddel(800), ndel
C    ------------------------------------
C          I N P U T   O F   FILE-NAMES
      type*, 'Name of input file of duplicate stations'
      accept 100, file1
  100 format(a12)
      open(unit=21, file=file1,status='old')
  333 continue
      read(21,111,end=112) nnn
      read(21,111) id
   25 format (2x,3i5)
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
      nm=0
      if(Depth(1).eq.0..or.depth(2).eq.0.)nm=1
      if(Depth(1).eq.0.)Depth(1)=0.000001
      if(Depth(2).eq.0.)Depth(2)=0.000001
      if(nm.eq.1)rewrite(21,52) Depth
C
      if(Depth(1).lt.0..and.Depth(2).lt.0.)goto 16
      go to 18
   52 format(2x,2f7.0)
   16 continue
   17 depth(1)=-1.*depth(1)
      depth(2)=-1.*depth(2)
      rewrite(21,52) Depth
   18 continue
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55)z(k),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2)
  111 format(2x,5i7)
      go to 333
  112 continue
      close(unit=21)
C     -----------------------------------------------------
      stop '*** E N D ***'
      end
```

Dup-10

```
      Program duplic8
C     Flagging of pairs of duplicate stations, which have been processed
C     Sequential number of pair of stations which has been processed
C     change its sign for negative
C
C     V.Guretsky, April, 1990, AWI
C     ------------------------------------
      real lon(2), lat(2), z(50), s(50,2),O2(50,2),t(50,2),
     *depth(2), modepth(2), dt(50),ds(50),dox(50)
      integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
      character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15, shipdel*15
      integer*4 nc(2), id(2), ncdel, idd1(800),idd2(800), ndel,m
C     ------------------------------------
      L=0
C          I N P U T   O F   FILE-NAMES
  100 format(a15)
      type*, 'Name of input file of duplicate stations'
      accept 100, file1
      open(unit=21, file=file1,status='old')
C
      type *, 'Name of input file of numbers of deleated stations '
      accept 100, file2
      open(unit=22, file=file2, status='old')
C
      type *,'name of input file of table of remaned stations'
      accept 100, file3
      open(unit=23, file=file3, status='old')
C          I N P U T   O F   D E L E A T E D   N U M B E R S
      jdel=1
   81 continue
      read(22,77, end=78) ndel,idd1(jdel),ncdel,shipdel
C
      read(23,111) m
      read(23,111)idd2(jdel)
      type*, jdel, idd1(jdel), idd2(jdel)
      read(23,50) m
      read(23,111) m
      read(23,51) a
      read(23,51) a
      read(23,52)a
      read(23,52)a
      read(23,111)m
      read(23,111)m
      read(23,111)m
      read(23,111)m
      read(23,111) m
      read(23,111) m
      read(23,111) n
      do 789 k=1,n
      read(23,96) a,b,c,d
  789 continue
c
      jdel=jdel+1
      goto 81
   78 continue
      type *, 'jdel=',jdel
   77 format(2x,3i7,a15)
C           INPUT OF PAIRS OF STATIONS
  333 continue
      mark=0
      read(21,111,end=112) nnn
      if(nnn.lt.0) go to 155
C       HERE THE PAIR IS MARKED IF IT HAS ALREADY BEEN PROCESSED
      do 15 k=1,jdel
```

Dup-5

```
      if(id(1).eq.idd1(k).and.id(2).eq.idd2(k)) mark = 1
      if(id(2).eq.idd1(k).and.id(1).eq.idd2(k)) mark = 1
   15 continue
   25 format (2x,3i5)
      if(mark.eq.0) go to 155
      nnn=-1*nnn
      rewrite(21,111) nnn
  155 continue
      read(21,111) id
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
   52 format(2x,2f7.0)
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55)z(k),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2)
  111 format(2x,5i7)
   96 format(2x,f5.0,1x,f8.3,1x,f8.3,1x,f6.2)
      go to 333
  112 continue
      close(unit=21)
      close(unit=22)
      close (unit=23)
C     -------------------------------------------------------
      stop '*** E N D ***'
      end
      Program duplic7
C     Correction of duplicate stations
C     V.Guretsky, May, 1990, AWI
C    -----------------------------------
      real lon(2), lat(2), z(50), s(50,2),O2(50,2),t(50,2),
     *depth(2), modepth(2), dt(50),ds(50),dox(50)
      integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
      character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15
      integer*4 nc(2), id(2)
C    -----------------------------------
      ncount=0
C          I N P U T
      type*, 'Name of input file'
      accept 100, file1
  100 format(a12)
      open(unit=21, file=file1,status='old')
      type *, 'Name of deleated stations file'
      accept 100, file2
      type*, 'Name of remained station file'
      accept 100, file3
  555 continue
      read(21,111,end=112) nnn
      read(21,111) id
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
   52 format(2x,2f7.0)
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55)z(k),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2)
      type*,nnn,n
  111 format(2x,5i7)
C     ---------------------------------------------------------
C          Criterium for the complete duplicates
      if(abs(dlon).ge.0.04) goto 555
      if(abs(dlat).ge.0.04) goto555
      do 2 k = 1,n
      if(dt(k).ge.0.005) go to 555
    2 if(ds(k).ge.0.005) go to 555
C     ---------------------------------------------------------
      jjj=1 ! keep
      mmm=2 ! deleate
      shipk=ship1
      shipd=ship2
      if(abs(numst(2)).lt.Abs(numst(1))) goto3
      goto4
    3 jjj=2 ! keep
      mmm=1 ! deleate
      shipd=ship1
      shipk=ship2
    4 continue
```

Dup-8

```
C     save remaining station at the place of the first duplicate
      id(1)=id(jjj)
      nc(1)=nc(jjj)
      ship1=shipk
      numst(1)=numst(jjj)
      lon(1)=(lon(1)+lon(2))/2.
      lat(1)=(lat(1)+lat(2))/2.
      depth(1)=depth(jjj)
      modepth(1)=modepth(jjj)
      nyear(1)=nyear(jjj)
      nmonth(1)=nmonth(jjj)
      nday(1)=nday(jjj)
      nhour(1)=max0(nhour(1),nhour(2))
      nob(1)=max0(nob(1),nob(2))
      do7 k = 1,n
      t(k,1)=(t(k,1)+t(k,2))/2.
      s(k,1)=(s(k,1)+s(k,2))/2.
      r=Abs(O2(k,1)-O2(k,2)) - 70.
      if(r)8,8,9
    8 O2(k,1)=amax1(O2(k,1),O2(k,2))
      goto 7
    9 O2(k,1)=amin1(O2(k,1),O2(k,2))
    7 continue
      ncount=ncount+1
      if(ncount.eq.1)go to 5
      goto6
    5 open(unit=22,file=file2,status='new')
      open(unit=23,file=file3,status='new')
    6 continue
C     ----------------------------------------------------
C                          O U T P U T
      write(22,200) ncount,id(mmm),nc(mmm), shipd
  200 format(2x,3i7,2x,a15)
      write(23,111) ncount
      write(23,111)id(1)
      write(23,50) nc(1), shipk
      write(23,111)numst(1)
      write(23,51)Lon(1)
      write(23,51)Lat(1)
      write(23,52)Depth(1)
      write(23,52)Modepth(1)
      write(23,111)nyear(1)
      write(23,111) nmonth(1)
      write(23,111) nday(1)
      write(23,111) nhour(1)
      write(23,111) nob(1)
      write(23,111)nms(1)
      write(23,111) n
      do11 k=1,n
   11 write(23,56) z(k),T(k,1),s(k,1),O2(k,1)
   56 format(2x,f5.0,1x,f8.3,1x,f8.3,1x,f6.2)
      goto555
  112 continue
      close(unit=21)
      close(unit=22)
      close(unit=23)
      stop '*** E N D ***'
      end
      PROGRAM DUPLIC6
C     SELECTION OF PAIRS OF DUPLICATE STATIONS AND WRITING THEM
C     V.GURETSKY, APREL 1990, A W I
C
      EXTERNAL                 err_handler
      EXTERNAL                 msg_handler
C
       include '(fsybdb)'
C
C     Variablendeklaration
C     --------------------
      Real z(50,2),t(50,2),s(50,2),O2(50,2),modepth(2),depth(2),
     *lon(2), lat(2)
      REAL*8 lon8,lat8,depth8,modepth8,z8,t8,s8,ox8
      CHARACTER cmdbuf*256, finp*12, fout*12, ship1*15, ship2*15
      INTEGER*2 nz(2),numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2),nms(2), numer
      INTEGER*4
     * login
     * ,dbproc
     * ,return_code
     * ,error
     *, stnum,year, month, day, hour, numobs, msq, id, nc,
     * id1,id2,nc1,nc2,nc3,nc4
C     --------------------------------------------
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
C     ----------------------------------------------
      login = fdblogin()
      call fdbsetluser(login, 'SOCEAN')
      call fdbsetlpwd(login, 'Victor')
      dbproc = fdbopen(login, NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
      call fdbsetnull(dbproc,flt8bind,0,99.)
C     ------------------------------------
C       I N P U T
      TYPE*, 'NAME OF INPUT FILE 12 CHARACTERS'
      read(6,100) finp
  100 format(a12)
      open(unit=20, file=finp, status='old')
        type*, 'Name of output file (12 CHARACTERS)'
        read(6,100)fout
        open(unit=21, file=fout,status='new')
  200 continue
      read(20,111,end=112) nnnn,id1,id2,nc1,nc2
      nc3=nc1
      nc4=nc2
      if(nc1.lt.0)nc3=30000-nc1
      if(nc2.lt.0)nc4=30000-nc2
C     --------------------
C      CALLS OF THE STORED PROCEDURE  Ship
      call fdbfcmd(dbproc,'Execute Ship %d', nc3)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,charbind,0,ship1)
      call fdbnextrow(dbproc)
      call fdbfcmd(dbproc,' Execute Ship %d',nc4)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,charbind,0,ship2)
      call fdbnextrow(dbproc)
C -------------------------
      Type*, 'num=',nnnn
  111 format(2x,5i7)
      id=id1
      do300 ii=1,2
```

Dup-7

```
      if(ii.eq.2)id=id2
      call fdbfcmd(dbproc,'Execute Dup3 %d', id)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,stnum)
      call fdbbind(dbproc,2,intbind,0,year)
      call fdbbind(dbproc,3,intbind,0,month)
      call fdbbind(dbproc,4,intbind,0,day)
      call fdbbind(dbproc,5,intbind,0,hour)
      call fdbbind(dbproc,6,flt8bind,0,lon8)
      call fdbbind(dbproc,7,flt8bind,0,lat8)
      call fdbbind(dbproc,8,flt8bind,0,depth8)
      call fdbbind(dbproc,9,flt8bind,0,Modepth8)
      call fdbbind(dbproc,10,intbind,0,numobs)
      call fdbbind(dbproc,11,intbind,0,msq)
      call fdbnextrow(dbproc)
C          Umwandlung von REAL*8 Variablen auf REAL
C          ----------------------------------------
           LON(ii)     = sngl(LON8)
           LAT(ii)     = sngl(LAT8)
           DEPTH(ii) = sngl(DEPTH8)
           MODEPTH(ii) = sngl(MODEPTH8)
           numst(ii)=stnum
           nyear(ii)=year
           nmonth(ii)=month
           nday(ii)=day
           nhour(ii)=hour
           nob(ii)=numobs
           nms(ii)=msq
C    ---------------------------------------------
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,intbind,0,n)
      call fdbnextrow(dbproc)
      nz(ii)=n
C    -----------------------------------------------
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,z8)
      call fdbbind(dbproc,2,flt8bind,0,t8)
      call fdbbind(dbproc,3,flt8bind,0,s8)
      call fdbbind(dbproc,4,flt8bind,0,ox8)
         j = 0
         do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
         J=J+1
         z(j,ii)=sngl(z8)
         t(j,ii)=sngl(t8)
         s(j,ii)=sngl(s8)
         O2(j,ii)=sngl(ox8)
         end do
  300 continue
C        REARRANGMENT OF TABLES
      N2=NZ(2)
      n1=nz(1)
      if(z(1,2)-z(1,1)) 70,76,71
   70 continue
C      Upper level of the first station is deeper
      m=0
      do 72 k=1,n2
      m=m+1
   72 if(z(1,1).eq.z(k,2)) go to 73
   73 k1=n1+m
      k2=n1+1
      do 74 k=1,n1
      z(k1-k,1) = z(k2-k,1)
      t(k1-k,1) = t(k2-k,1)
      s(k1-1,1)=  s(k2-k,1)
   74 O2(k1-k,1)=O2(k2-k,1)
```

```
      do 75 k=1,m
      z(k,1)=99.
      t(k,1)=99.
      s(k,1)=99.
   75 O2(k,1)=99.
      go to 76
   71 continue
C     Upper level of the second station is deeper
      m=0
      do 92 k=1,n1
      m=m+1
   92 if(z(1,2).eq.z(k,1)) go to 93
   93 k1=n2+m
      k2=n2+1
      do 94 k=1,n2
      z(k1-k,2)=z(k2-k,2)
      t(k1-k,2)=t(k2-k,2)
      s(k1-k,2)=s(k2-k,2)
   94 O2(k1-k,2)=O2(k2-k,2)
      do 95 k=1,m
      z(k,2)=99.
      t(k,2)=99.
      s(k,2)=99.
   95 O2(k,2)=99.
   76 continue
      n=imax0(nz(1),nz(2))
C     ********************************
C               O U T P U T
      write(21,111)nnnn
      write(21,111)id1,id2
      write(21,50) nc1,ship1,nc2,ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      nn=numst(2)-numst(1)
      write(21,111) numst,nn
      dd=lon(2)-lon(1)
      write(21,51) Lon, dd
      dd=Lat(2)-Lat(1)
      write(21,51) Lat, dd
   51 format(2x,3f8.3)
      write(21,52) Depth
   52 format(2x,2f7.0)
      write(21,52) Modepth
      write(21,111) nyear
      write(21,111) nmonth
      write(21,111) nday
      nn=nhour(2)-nhour(1)
      write(21,111) nhour,nn
      nn=nob(2)-nob(1)
      write(21,111) nob,nn
      nn=nms(2)-nms(1)
      write(21,111) nms,nn
      write(21,111) n
      do 27 k=1,n
      tt=t(k,2)-t(k,1)
      ss=s(k,2)-s(k,1)
      xx=O2(k,2)-O2(k,1)
   27 write(21,55) z(k,1),(t(k,j),j=1,2),tt,(s(k,j),j=1,2),ss,
     *(O2(k,j),j=1,2),xx
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2)
      go to 200
C   ***********************************
  112 continue
      close(unit=21)
      call fdbexit()                         ! Schliessen der DB-Library
C
      CLOSE(UNIT=20)
      stop '********* E N D *********'
      END
C
C -----------------------------------------

C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
C      EXTERNAL         err_handler
C      EXTERNAL         msg_handler
C
      INTEGER*4       dbproc
      INTEGER*4       severity
      INTEGER*4       errno
      INTEGER*4       oserrno
      INTEGER*4       length
      INTEGER*4       return_code
C
      CHARACTER*(80)  message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C      Message Handler
C      ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2              msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4      dbproc
      INTEGER*4       msgno
      INTEGER*4       msgstate
      INTEGER*4       severity
C
      CHARACTER*80    msgtext
        IF (MSGNO.NE.5701) THEN
C
          type *, 'DataServer message ', msgno,
     2       '    state ', msgstate, '     severity ',
     3       severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
```

June 90

```
                   Program duplic97
C    all pairs having positive sequential number
C    are typed
C                   V.Guretsky, May, 1990, AWI
C    -----------------------------------
          Real lon(2), lat(2), z(50,2), s(50,3),O2(50,3),t(50,3),
     *depth(2), modepth(2), dt(50),ds(50),dox(50),lonr, latr, modepthr,
     *depthr
C
          Integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
C
          Character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15, X*1
C
          Integer*4 nc(2), id(2),ncr, idr, jjjj,nstc1,nstc2
C     -  -----------------------------------
C-----------------------------------
      ncount=0
      icount=0
      ncount1=0
C          I N P U T
      type*, 'Name of input file of pairs of stations'
      accept 100, file1
  100 format(a15)
      open(unit=21, file=file1,status='old')
C
      type *, 'Name of outputfile of stations to delete'
      accept 100, file2
      open(unit=22,file=file2,status='new')
C
      type*,'name of outputfile for nonduplicates id'
      accept 100, file3
      open(unit=23,file=file3, status='new')
C         ----------------------------------------------------
  555 continue
C     **************************************************
      read(21,111,end=112) nnn, nstc1, nstc2
      if(nnn.le.0)icount=icount+1
      read(21,111) id
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
   52 format(2x,2f7.0)
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55) z(k,1),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k),z(k,2)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2,1x,f5.0)
  515 format(2x,f5.0,1x,3f7.3,1x,3f7.3,1x,3f6.2,1x,f5.0)
C***********************************************************
C        CHECK IF THIS PAIR HAS ALREADY BEEN PROCESSED
      if(nnn.le.0) goto 555
  111 format(2x,5i7)
C     -----------------------------------------------------
```

Dup-b

```
C           T Y P E   S T A T I O N S   O N   T H E   S C R E E N
  444 continue
      type 111, nnn, nstc1, nstc2
      type 111, id
      type 50, nc(1),Ship1, nc(2),ship2
      type 111, numst
      type 51, Lon, dlon
      type 51, Lat, dlat
      type 52, Depth
      type 52, Modepth
      type 111, nyear
      type 111, nmonth
      type 111, nday
      type 111, nhour
      type 111, nob
      type 111, nms
      type 111, n
C     -----------------------------
  156 format(a1)
C     -----------------------------
      do 28 k = 1, n
   28 type 515, z(k,1), (t(k,j),j=1,2), dt(k), (s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k),z(k,2)
C     -----------------------------
      type*,'$$$$$  type station again? 0 - no  1 - yes'
      accept 57,k
      if(k)445,445,444
   57 format(2i1)
  445 continue
C-------------------------------------------------------------------------------
C        W H I C H   S T A T I O N   T O   KEEP
      type*,'$$$$$ (KEEP 1 or 2) (NON DUPLIC 3) (NO selec > 3)'
      accept 57,jjj
C    ---------------------------------------
      if (jjj.gt.3)go to 555
      if(jjj.eq.3) go to 556
C    --------------------------
      if(jjj.eq.1) mmm=2
      if(jjj.eq.2) mmm=1
      if(jjj-1) 43,43,44
   43 shipk=ship1
      shipd=ship2
      go to 45
   44 continue
      shipk=ship2
      shipd=ship1
   45 continue
      idr=id(jjj)
      ncr=nc(jjj)
      lonr=lon(jjj)
      latr=lat(jjj)
      numstr=numst(jjj)
      depthr=depth(jjj)
      modepthr=modepth(jjj)
      nyearr=nyear(jjj)
      nmonthr=nmonth(jjj)
      ndayr=nday(jjj)
      nhourr=nhour(jjj)
      nobr=nob(jjj)
C     -----------------
      ncount=ncount+1
C     -------------------------------------------------
C                          O U T P U T
      write(22,200) ncount,id(mmm),nc(mmm), shipd, id(jjj), nc(jjj),
     *shipk
      go to 555
  556 ncount1=ncount1+1
      write(23,200)ncount1,id(1),nc(1),ship1,id(2),nc(2),ship2
C  -----------------------------------------------
  200 format(2x,3i7,2x,a15,2x,2i7,2x,a15)
   56 format(2x,f5.0,1x,f8.3,1x,f8.3,1x,f6.2)
C
      goto555
  112 continue
      np=100*ncount/nnn
      Type*,ncount,'  stations passed this test'
      type*,np,' prcents'
      nsum=icount+ncount
      np=100*nsum/nnn
      type*,nsum,' stations are processed alltogether'
      type*,np,' percents'
      close(unit=21)
      close(unit=22)
      close(unit=23)
      stop '*** E N D ***'
      end
```

June 90

```
                        Program duplic96
C     Correction of duplicate stations (test 6)
C         Pairs having
C     not less then 50 percent of levels where dT and dS
C     are less or equal 0.004
C     are considered to be possible  duplicates.
C         Out of them one file is  constructed, e.g. file of numbers
C     of stations to be deleated
C                     V.Guretsky, May, 1990, AWI
C     -----------------------------------
          Real lon(2), lat(2), z(50,2), s(50,3),O2(50,3),t(50,3),
     *depth(2), modepth(2), dt(50),ds(50),dox(50),lonr, latr, modepthr,
     *depthr
C
          Integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
C
          Character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15, X*1
C
          Integer*4 nc(2), id(2),ncr, idr, jjjj,nstc1,nstc2
C      -  ----------------------------------
C------------------------------------
      ncount=0
      ncount1=0
      icount=0
C          I N P U T
      type*, 'Name of input file of pairs of stations'
      accept 100, file1
  100 format(a15)
      open(unit=21, file=file1,status='old')
C
      type *, 'Name of outputfile of stations to delete'
      accept 100, file2
      open(unit=22,file=file2,status='new')
C
      type*,'Name of output file of nonduplicate Id'
      accept 100, file3
      open(unit=23,file=file3,status='new')
C         -------------------------------------------------------
  555 continue
C     **************************************************
      read(21,111,end=112) nnn, nstc1, nstc2
      if(nnn.le.0)icount=icount+1
      read(21,111) id
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
   52 format(2x,2f7.0)
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55) z(k,1),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k),z(k,2)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2,1x,f5.0)
  515 format(2x,f5.0,1x,3f7.3,1x,3f7.3,1x,3f6.2,1x,f5.0)
```

Du-55

```
C*****************************************************************
C         CHECK IF THIS PAIR HAS ALREADY BEEN PROCESSED
      if(nnn.le.0) goto 555
  111 format(2x,5i7)
C      -------------------------------------------------------
C            T and S criterium for duplicates
      mt=0
      ms=0
      do 2 k = 1,n
      if(abs(dt(k)).ge.0.005) mt=mt+1
    2 if(abs(ds(k)).ge.0.005) ms=ms+1
      mtp=mt*100/n
      msp=ms*100/n
      if(mtp.ge.50.or.msp.ge.50) go to 555
C             T Y P E  S T A T I O N S   O N   T H E   S C R E E N
  444 continue
      type 111, nnn, nstc1, nstc2
      type 111, id
      type 50, nc(1),Ship1, nc(2),ship2
      type 111, numst
      type 51, Lon, dlon
      type 51, Lat, dlat
      type 52, Depth
      type 52, Modepth
      type 111, nyear
      type 111, nmonth
      type 111, nday
      type 111, nhour
      type 111, nob
      type 111, nms
      type 111, n
C     ---------------------------
  156 format(a1)
C     ---------------------------
      do 28 k = 1, n
   28 type 515, z(k,1), (t(k,j),j=1,2), dt(k), (s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k),z(k,2)
C     ---------------------------
      type*,'$$$$$  type station again? 0 - no  1 - yes'
      accept 57,k
      if(k)445,445,444
   57 format(i1)
  445 continue
C--------------------------------------------------------------------------
      type*,'$$$$$ Type: (KEEP: 1 or 2)  (DIFFER 3)  (NO selection >3)'
      accept 57,jjj
C    --------------------------------------
      if (jjj.eq.3) go to 557
      if(jjj.gt.3) go to 555
C    --------------------------
      if(jjj.eq.1) mmm=2
      if(jjj.eq.2) mmm=1
      if(jjj-1) 43,43,44
   43 shipk=ship1
      shipd=ship2
      go to 45
   44 continue
      shipk=ship2
      shipd=ship1
   45 continue
      idr=id(jjj)
      ncr=nc(jjj)
      lonr=lon(jjj)
      latr=lat(jjj)
      numstr=numst(jjj)
      depthr=depth(jjj)
      modepthr=modepth(jjj)
      nyearr=nyear(jjj)
      nmonthr=nmonth(jjj)
      ndayr=nday(jjj)
      nhourr=nhour(jjj)
      nobr=nob(jjj)
C     -----------------
  556 continue
      ncount=ncount+1
C     ---------------------------------------------------
C                         O U T P U T
       write(22,200) ncount,id(mmm),nc(mmm), shipd, id(jjj), nc(jjj),
     * shipk
       go to 555
  557 continue
      ncount1=ncount1+1
      write(23,200) ncount1,id(1),nc(1),ship1,id(2),nc(2),ship2
C  ---------------------------------------------------
  200 format(2x,3i7,2x,a15,2x,2i7,2x,a15)
      goto555
C
  112 continue
      np=100*ncount/nnn
      Type*,ncount,'  stations passed this test'
      type*,np,' prcents'
      nsum=icount+ncount
      np=100*nsum/nnn
      type*,nsum,' stations are processed alltogether'
      type*,np,' percents'
      close(unit=21)
      close(unit=22)
      close(unit=23)
      stop '*** E N D ***'
      end
```

June 90

```
                    Program duplic94
C     Correction of duplicate stations (test 4)
C         Pairs having coordinate difference less than 0.1 degree
C     are considered to be possible duplicates.
C         Out of them one file is  constructed, e.g. file of numbers
C     of stations to be deleated
C                    V.Guretsky, May, 1990, AWI
C    -------------------------------------
          Real lon(2), lat(2), z(50,2), s(50,3),O2(50,3),t(50,3),
     *depth(2), modepth(2), dt(50),ds(50),dox(50),lonr, latr, modepthr,
     *depthr
C
          Integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
C
          Character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15, X*1
C
          Integer*4 nc(2), id(2),ncr, idr, jjjj,nstc1,nstc2
C     -  ---------------------------------
      ncount=0
      mcount=0
C          I N P U T
      type*, 'Name of input file of pairs of stations'
      accept 100, file1
  100 format(a15)
      open(unit=21, file=file1,status='old')
C
      type *, 'Name of outputfile of stations to delete'
      accept 100, file2
      open(unit=22,file=file2,status='new')
C
      type*,'Name of outputfile of stations to delete'
      accept 100, file3
      open(unit=23, file=file3, status='new')
C        -------------------------------------------------------
  555 continue
C     ************************************************
      read(21,111,end=112) nnn, nstc1, nstc2
C          if(nnn.gt.30) go to 112
      read(21,111) id
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
   52 format(2x,2f7.0)
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55)z(k,1),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k),z(k,2)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2,1x,f5.0)
  515 format(1x,f5.0,3f8.3,1x,2f7.3,f8.3,1x,3f6.2,1x,f5.0)
C*****************************************************************
      type*,nnn
C        CHECK IF THIS PAIR HAS ALREADY BEEN PROCESSED
      if(nnn.lt.0) goto 555
```

Dup-4

```
  111 format(2x,5i7)
C       -------------------------------------------------------
C           T and S criterium for duplicates
C        do 2 k = 1,n
C        if(abs(dt(k)).ge.0.005) go to 555
C      2 if(abs(ds(k)).ge.0.005) go to 555
C        -------------------------------------------------------
C           Criterium for coordinates
        if(abs(dlon).ge.0.1) go to 555
        if(abs(dlat).ge.0.1) go to 555
C              T Y P E   S T A T I O N S   O N   T H E   S C R E E N
  444 continue
        type 111, nnn, nstc1, nstc2
        type 111, id
        type 50, nc(1),Ship1, nc(2),ship2
        type 111, numst
        type 51, Lon, dlon
        type 51, Lat, dlat
        type 52, Depth
        type 52, Modepth
        type 111, nyear
        type 111, nmonth
        type 111, nday
        type 111, nhour
        type 111, nob
        type 111, nms
        type 111, n
C       -----------------------------
  156 format(a1)
C        ----------------------------
        do 28 k = 1, n
   28 type 515, z(k,1), (t(k,j),j=1,2), dt(k), (s(k,j),j=1,2),ds(k),
      *(O2(k,j),j=1,2),dox(k),z(k,2)
C        ---------------------------
        type*,'$$$$$  type station again? 0 - no  1 - yes'
        accept 57,kk
        if(kk)445,445,444
   57 format(2i1)
  445 continue
C------------------------------------------------------------------------------
C         W H I C H   S T A T I O N   T O    KEEP
        type*,'$$$$$ KEEP 1 or 2;  nonduplicates 3; nooutput >3'
        accept 57,jjj
C      ----------------------------------------
        if (jjj.gt.3) go to 555
        if(jjj.gt.3) go to 655
C      ---------------------------
        if(jjj.eq.1) mmm=2
        if(jjj.eq.2) mmm=1
        if(jjj-1) 43,43,44
   43 shipk=ship1
        shipd=ship2
        go to 45
   44 continue
        shipk=ship2
        shipd=ship1
   45 continue
        idr=id(jjj)
        ncr=nc(jjj)
        lonr=lon(jjj)
        latr=lat(jjj)
        numstr=numst(jjj)
        depthr=depth(jjj)
        modepthr=modepth(jjj)
        nyearr=nyear(jjj)
        nmonthr=nmonth(jjj)
      ndayr=nday(jjj)
      nhourr=nhour(jjj)
      nobr=nob(jjj)
C     -----------------
      ncount=ncount+1
C     --------------------------------------------------
C                           O U T P U T
      write(22,200) ncount,id(mmm),nc(mmm), shipd, id(jjj), nc(jjj),
     *shipk
       go to 555
  655 continue
      mcount=mcount+1
      write(23,200) mcount,id(1),nc(1),ship1,id(2),nc(2),ship2
C ---------------------------------------------------
  200 format(2x,3i7,2x,a15,2x,2i7,2x,a15)
C      write(23,111) ncount
C      write(23,111)idr
C      write(23,50) ncr, shipk
C      write(23,111)numstr
C      write(23,51)Lonr
C      write(23,51)Latr
C      write(23,52)Depthr
C      write(23,52)Modepthr
C      write(23,111)nyearr
C      write(23,111) nmonthr
C      write(23,111) ndayr
C      write(23,111) nhourr
C      write(23,111) nobr
C      write(23,111)nmsr
C      write(23,111) n
C      do11 k=1,n
C   11 write(23,56) z(k,jjj),T(k,3),s(k,3),O2(k,3)
   56 format(2x,f5.0,1x,f8.3,1x,f8.3,1x,f6.2)
C
      goto555
  112 continue
      close(unit=21)
      close(unit=22)
      close(unit=23)
      stop '*** E N D ***'
      end
```

June 90

```
                          Program duplic93
C       Correction of duplicate stations (test 3)
C          Pairs having t or S difference le 0.004
C       are considered to be possible duplicates.
C          Out of them one file is  constructed, e.g. file of numbers
C       of stations to be deleated
C                     V.Guretsky, May, 1990, AWI
C      ---------------------------------
           Real lon(2), lat(2), z(50,2), s(50,3),O2(50,3),t(50,3),
      *depth(2), modepth(2), dt(50),ds(50),dox(50),lonr, latr, modepthr,
      *depthr
C
           Integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
      *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
C
           Character file1*15, file2*15, file3*15, ship1*15, ship2*15,
      *shipd*15, shipk*15, X*1
C
           Integer*4 nc(2), id(2),ncr, idr, jjjj,nstc1,nstc2
C       -  ---------------------------------
       ncount=0
C           I N P U T
       type*, 'Name of input file of pairs of stations'
       accept 100, file1
  100  format(a15)
       open(unit=21, file=file1,status='old')
C
       type *, 'Name of outputfile of stations to delete'
       accept 100, file2
       open(unit=22,file=file2,status='new')
C      -------------------------------------------------------
  555  continue
C      ***********************************************
       read(21,111,end=112) nnn, nstc1, nstc2
C            if(nnn.gt.30) go to 112
       read(21,111) id
       read(21,50) nc(1),Ship1,nc(2),Ship2
   50  format(2x,i7,2x,a15,2x,i7,2x,a15)
       read(21,111)(numst(j),j=1,2)
       read(21,51) Lon(1),Lon(2),dlon
       read(21,51) Lat(1),Lat(2),dlat
   51  format(2x,3f8.3)
       read(21,52)Depth
   52  format(2x,2f7.0)
       read(21,52)Modepth
       read(21,111)nyear
       read(21,111)nmonth
       read(21,111)nday
       read(21,111)nhour
       read(21,111)nob
       read(21,111)nms
       read(21,111)n
       do 27 k = 1, n
   27  read(21,55)z(k,1),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
      *(O2(k,j),j=1,2),dox(k),z(k,2)
   55  format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2,1x,f5.0)
  515  format(1x,f5.0,1x,3f7.3,1x,3f7.3,1x,3f6.2,1x,f5.0)
C*************************************************************
       type*,nnn
C         CHECK IF THIS PAIR HAS ALREADY BEEN PROCESSED
       if(nnn.lt.0) goto 555
  111  format(2x,5i7)
C       ----------------------------------------------------
C            T and S criterium for duplicates
       do 2 k = 1,n
       if(abs(dt(k)).ge.0.005) go to 555
```

Dup-2

```
    2 if(abs(ds(k)).ge.0.005) go to 555
C      -------------------------------------------------------
C          Criterium for coordinates
CC      if(abs(dlon).ge.0.1) go to 555
CC      if(abs(dlat).ge.0.1) go to 555
C             T Y P E   S T A T I O N S   O N   T H E   S C R E E N
  444 continue
      type 111, nnn, nstc1, nstc2
      type 111, id
      type 50, nc(1),Ship1, nc(2),ship2
      type 111, numst
      type 51, Lon, dlon
      type 51, Lat, dlat
      type 52, Depth
      type 52, Modepth
      type 111, nyear
      type 111, nmonth
      type 111, nday
      type 111, nhour
      type 111, nob
      type 111, nms
      type 111, n
C     -----------------------------
  156 format(a1)
C      ----------------------------
      do 28 k = 1, n
   28 type 515, z(k,1), (t(k,j),j=1,2), dt(k), (s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k),z(k,2)
C      ----------------------------
      type*,'$$$$$ type station again? 0 - no  1 - yes'
      accept 57,k
      if(k)445,445,444
   57 format(2i1)
  445 continue
C-----------------------------------------------------------------------
C         W H I C H  S T A T I O N   T O   KEEP
      type*,'$$$$$ Type which station to keep  IF TYPE 3 NO OUTPUT '
      accept 57,jjj
C     ---------------------------------------
      if (jjj.gt.2) go to 555
C     --------------------------
      if(jjj.eq.1) mmm=2
      if(jjj.eq.2) mmm=1
      if(jjj-1) 43,43,44
   43 shipk=ship1
      shipd=ship2
      go to 45
   44 continue
      shipk=ship2
      shipd=ship1
   45 continue
      idr=id(jjj)
      ncr=nc(jjj)
      lonr=lon(jjj)
      latr=lat(jjj)
      numstr=numst(jjj)
      depthr=depth(jjj)
      modepthr=modepth(jjj)
      nyearr=nyear(jjj)
      nmonthr=nmonth(jjj)
      ndayr=nday(jjj)
      nhourr=nhour(jjj)
      nobr=nob(jjj)
C     -----------------
      ncount=ncount+1
C     -------------------------------------------------
```

```
C                        O U T P U T
       write(22,200) ncount,id(mmm),nc(mmm), shipd, id(jjj), nc(jjj),
      *shipk
C ----------------------------------------------------
  200 format(2x,3i7,2x,a15,2x,2i7,2x,a15)
C        write(23,111) ncount
C        write(23,111)idr
C        write(23,50) ncr, shipk
C        write(23,111)numstr
C        write(23,51)Lonr
C        write(23,51)Latr
C        write(23,52)Depthr
C        write(23,52)Modepthr
C        write(23,111)nyearr
C        write(23,111) nmonthr
C        write(23,111) ndayr
C        write(23,111) nhourr
C        write(23,111) nobr
C        write(23,111)nmsr
C        write(23,111) n
C        do11 k=1,n
C     11 write(23,56) z(k,jjj),T(k,3),s(k,3),O2(k,3)
   56 format(2x,f5.0,1x,f8.3,1x,f8.3,1x,f6.2)
C
      goto555
  112 continue
      close(unit=21)
      close(unit=22)
      stop '*** E N D ***'
      end
```

June 90

```
                    Program duplic92
C     Correction of duplicate stations (test 2)
C         Pairs having  coordinates difference not more than 0.1 degree
C     and having not less then 50 percent of levels where dT and dS
C     are less or equal 0.004
C     are considered to be possible  duplicates.
C         Out of them one file is  constructed, e.g. file of numbers
C     of stations to be deleated
C                    V.Guretsky, May, 1990, AWI
C    -------------------------------------
          Real lon(2), lat(2), z(50,2), s(50,3),O2(50,3),t(50,3),
     *depth(2), modepth(2), dt(50),ds(50),dox(50),lonr, latr, modepthr,
     *depthr
C
          Integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
C
          Character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15, X*1
C
          Integer*4 nc(2), id(2),ncr, idr, jjjj,nstc1,nstc2
C    -  ----------------------------------
C------------------------------------
      ncount=0
      mcount=0
      icount=0
C         I N P U T
      type*, 'Name of input file of pairs of stations'
      accept 100, file1
  100 format(a15)
      open(unit=21, file=file1,status='old')
C
      type *, 'Name of outputfile of stations to delete'
      accept 100, file2
      open(unit=22,file=file2,status='new')
      type *, 'Name of output file of nonduplicates'
      accept 100, file3
      open(unit=23, file=file3,status='new')
C     ---------------------------------------------------------
  555 continue
C     ***********************************************
      read(21,111,end=112) nnn, nstc1, nstc2
      type*,nnn
      if(nnn.le.0)icount=icount+1
      read(21,111) id
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
   52 format(2x,2f7.0)
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55) z(k,1),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k),z(k,2)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2,1x,f5.0)
  515 format(2x,f5.0,1x,3f7.3,1x,3f7.3,1x,3f6.2,1x,f5.0)
```

Dup-2

```
C*****************************************************************
C        CHECK IF THIS PAIR HAS ALREADY BEEN PROCESSED
      if(nnn.le.0) goto 555
  111 format(2x,5i7)
C      -------------------------------------------------------
C          T and S criterium for duplicates
      mt=0
      ms=0
      do 2 k = 1,n
      if(abs(dt(k)).ge.0.005) mt=mt+1
    2 if(abs(ds(k)).ge.0.005) ms=ms+1
      mtp=mt*100/n
      msp=ms*100/n
      if(mtp.ge.50.or.msp.ge.50) go to 555
C
C      -------------------------------------------------------
C          Criterium for coordinates
      if(abs(dlon).ge.0.1) go to 555
      if(abs(dlat).ge.0.1) go to 555
C             T Y P E   S T A T I O N S   O N   T H E   S C R E E N
  444 continue
      type 111, nnn, nstc1, nstc2
      type 111, id
      type 50, nc(1),Ship1, nc(2),ship2
      type 111, numst
      type 51, Lon, dlon
      type 51, Lat, dlat
      type 52, Depth
      type 52, Modepth
      type 111, nyear
      type 111, nmonth
      type 111, nday
      type 111, nhour
      type 111, nob
      type 111, nms
      type 111, n
C     -----------------------------
  156 format(a1)
C      ----------------------------
      do 28 k = 1, n
   28 type 515, z(k,1), (t(k,j),j=1,2), dt(k), (s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k),z(k,2)
C      ----------------------------
      type*,'$$$$$  type station again? 0 - no  1 - yes'
      accept 57,k
      if(k)445,445,444
   57 format(2i1)
  445 continue
C---------------------------------------------------------------------------
C        W H I C H   S T A T I O N   T O   KEEP
      type*,'$$$$$ KEEP 1 or 2;  3-nonduplicates; >3  NO OUTPUT'
      accept 57,jjj
C    ---------------------------------------
      if (jjj.gt.3) go to 555
      if(jjj.eq.3) go to 655
C    --------------------------
      if(jjj.eq.1) mmm=2
      if(jjj.eq.2) mmm=1
      if(jjj-1) 43,43,44
   43 shipk=ship1
      shipd=ship2
      go to 45
   44 continue
      shipk=ship2
      shipd=ship1
   45 continue
```

```
      idr=id(jjj)
      ncr=nc(jjj)
      lonr=lon(jjj)
      latr=lat(jjj)
      numstr=numst(jjj)
      depthr=depth(jjj)
      modepthr=modepth(jjj)
      nyearr=nyear(jjj)
      nmonthr=nmonth(jjj)
      ndayr=nday(jjj)
      nhourr=nhour(jjj)
      nobr=nob(jjj)
C     -----------------
      ncount=ncount+1
C     --------------------------------------------------
C                        O U T P U T
      write(22,200) ncount,id(mmm),nc(mmm), shipd, id(jjj), nc(jjj),
     *shipk
      go to 555
  655 continue
      mcount=mcount+1
      write(23,200)mcount,id(1),nc(1),ship1,id(2),nc(2),ship2
      go to 555
C  ------------------------------------------------
  200 format(2x,3i7,2x,a15,2x,2i7,2x,a15)
C      write(23,111) ncount
C      write(23,111)idr
C      write(23,50) ncr, shipk
C      write(23,111)numstr
C      write(23,51)Lonr
C      write(23,51)Latr
C      write(23,52)Depthr
C      write(23,52)Modepthr
C      write(23,111)nyearr
C      write(23,111) nmonthr
C      write(23,111) ndayr
C      write(23,111) nhourr
C      write(23,111) nobr
C      write(23,111)nmsr
C      write(23,111) n
C      do11 k=1,n
C   11 write(23,56) z(k,jjj),T(k,3),s(k,3),O2(k,3)
   56 format(2x,f5.0,1x,f8.3,1x,f8.3,1x,f6.2)
C
  112 continue
      np=100*ncount/nnn
      Type*,ncount,'  stations passed this test'
      type*,np,' prcents'
      nsum=icount+ncount
      np=100*nsum/nnn
      type*,nsum,' stations are processed alltogether'
      type*,np,' percents'
      close(unit=21)
      close(unit=22)
      close(unit=23)
      stop '*** E N D ***'
      end
```

June 90

```
                         Program duplic91
C     Correction of duplicate stations (test 1)
C         Pairs having the same T and S data (within 0.005 plus-minus)
C     and coordinates difference not more than 0.1 degree are
C     considered to be exact duplicates.
C         Out of them one file is  constructed, e.g. file of numbers
C     of stations to be deleated
C                    V.Guretsky, May, 1990, AWI
C    -------------------------------------
          Real lon(2), lat(2), z(50,2), s(50,3),O2(50,3),t(50,3),
     *depth(2), modepth(2), dt(50),ds(50),dox(50),lonr, latr, modepthr,
     *depthr
C
          Integer*2 numst(2), nyear(2), nmonth(2), nday(2), nhour(2),
     *nob(2), nms(2), numer,nnn,n,nhourd,nobsd,nmsd
C
          Character file1*15, file2*15, file3*15, ship1*15, ship2*15,
     *shipd*15, shipk*15
C
          Integer*4 nc(2), id(2),ncr, idr, jjjj,nstc1,nstc2
C     -  -----------------------------------
      tmin=-2.3
      tmax=30.0
      smin1=27.
      smin2=33.5
      smax=35.2
C-----------------------------------
      ncount=0
C          I N P U T
      type*, 'Name of input file of pairs of stations'
      accept 100, file1
  100 format(a12)
      open(unit=21, file=file1,status='old')
      type *, 'Name of output file of numbers of stations to delete'
      accept 100, file2
C            type*, 'Name of remained station file'
C            accept 100, file3
C        --------------------------------------------------------
  555 continue
C        ***********************************************
      read(21,111,end=112) nnn, nstc1, nstc2
      read(21,111) id
      read(21,50) nc(1),Ship1,nc(2),Ship2
   50 format(2x,i7,2x,a15,2x,i7,2x,a15)
      read(21,111)(numst(j),j=1,2)
      read(21,51) Lon(1),Lon(2),dlon
      read(21,51) Lat(1),Lat(2),dlat
   51 format(2x,3f8.3)
      read(21,52)Depth
   52 format(2x,2f7.0)
      read(21,52)Modepth
      read(21,111)nyear
      read(21,111)nmonth
      read(21,111)nday
      read(21,111)nhour
      read(21,111)nob
      read(21,111)nms
      read(21,111)n
      do 27 k = 1, n
   27 read(21,55)z(k,1),(t(k,j),j=1,2),dt(k),(s(k,j),j=1,2),ds(k),
     *(O2(k,j),j=1,2),dox(k),z(k,2)
   55 format(2x,f5.0,1x,3f8.3,1x,3f8.3,1x,3f6.2,1x,f5.0)
C*************************************************************
C          CHECK IF THIS PAIR HAS ALREADY BEEN PROCESSED
      if(nnn.lt.0) goto 555
  111 format(2x,5i7)
```

Dup -1

```
C       ----------------------------------------------------------
C            T and S criterium for duplicates
        do 2 k = 1,n
        if(abs(dt(k)).ge.0.005) go to 555
      2 if(abs(ds(k)).ge.0.005) go to 555
C       ----------------------------------------------------------
C            Criterium for coordinates
        if(abs(dlon).ge.0.1) go to 555
        if(abs(dlat).ge.0.1) go to 555
        type111,nnn,nstc1,nstc2
        type111,id
C     ----------------------------------------
C       *1*    Here we determine which station to keep using oxygen data
C           We take station with oxygen. If both have the same oxygen
C            data(even dummy) no decision is made
        j1=0
        j2=0
        do7 k = 1,n
        t(k,3)=(t(k,1)+t(k,2))/2.
        s(k,3)=(s(k,1)+s(k,2))/2.
        r=Abs(O2(k,1)-O2(k,2)) - 70.
        if(r)8,8,9
      8 O2(k,3)=amax1(O2(k,1),O2(k,2))
        goto 71
      9 O2(k,3)=amin1(O2(k,1),O2(k,2))
     71 continue
        if(O2(k,3).eq.O2(k,1)) j1=j1+1
        if(O2(k,3).eq.O2(k,2)) j2=j2+1
      7 continue
        L=0
        do 78 k=1,n
        y=Abs(O2(k,1)-O2(k,2))
        if(y.gt.0.02) L=1
     78 continue
        if(L.eq.0) go to 75  ! decision is not made
        if(j1-j2) 72,75,74
     72 jjj=2
        mmm=1
        ntest=1
        go to 76
     74 jjj=1
        mmm=2
        ntest=1
        go to 76
C       ------------------------------------------------------------------
     75 continue
C
C       *2*  Here  we determine which station to keep using Station_Number
C       We keep station with positive and less then 500 Station number
C
        if(abs(numst(1))-abs(numst(2))) 82,85,84
     82 jjj=1
        mmm=2
        ntest=2
        go to 76
     84 jjj=2
        mmm=1
        ntest=2
        go to 76
C     -----------------------------------------------------------
     85 continue
C
C       *3*    Here we determine which station to keep using Number_Obs
C       We keep station with greater number of observations
C
C       if(nob(1)-nob(2)) 91, 95, 93
```

```
   91 jjj=2
      mmm=1
      ntest=3
      go to 76
   93 jjj=1
      mmm=2
      ntest=3
      go to 76
   95 continue
C-----------------------------------------------------------------
C     *4*  Here we made decision by comparing nimber of stations for
C     the first and second cruises of the pair
      if(nstc1-nstc2) 60,63,62
   60 jjj=2
      mmm=1
      ntest=4
      go to 76
   62 jjj=1
      mmm=2
      ntest=4
      go to 76
   63 continue
C
C----------------------------------------------------------------------------
C     *5*  Here we made decision which station to keep by Station_Id
C      Station with greater ID is deleted
      jjjj=id(1)-id(2)
      if(jjjj) 41,41,42
   41 jjj=1
      mmm=2
      ntest=5
      go to 76
   42 jjj=2
      mmm=1
      ntest=5
C
C----------------------------------------------------------------------
C       Here we got all information for stations to keep and to delete
C       (but in this program we need in fact only information for the
C         stations to be deleted, e.g. having index  mmm )
   76 continue
      if(jjj-1) 43,43,44
   43 shipk=ship1
      shipd=ship2
      go to 45
   44 continue
      shipk=ship2
      shipd=ship1
   45 continue
      idr=id(jjj)
      ncr=nc(jjj)
      lonr=lon(jjj)
      latr=lat(jjj)
      numstr=numst(jjj)
      depthr=depth(jjj)
      modepthr=modepth(jjj)
      nyearr=nyear(jjj)
      nmonthr=nmonth(jjj)
      ndayr=nday(jjj)
      nhourr=nhour(jjj)
      nobr=nob(jjj)
      ncount=ncount+1
      if(ncount.eq.1)go to 5
      goto6
    5 open(unit=22,file=file2,status='new')
C                     open(unit=23,file=file3,status='new')
```

```
    6 continue
C     -------------------------------------------------
C                        O U T P U T
      write(22,200) ncount,id(mmm),nc(mmm), shipd, id(jjj), nc(jjj),
     *shipk
C      type*, ncount, id(mmm),id(jjj),ntest
  200 format(2x,3i7,2x,a15,2x,2i7,2x,a15)
C      write(23,111) ncount
C      write(23,111)idr
C      write(23,50) ncr, shipk
C      write(23,111)numstr
C      write(23,51)Lonr
C      write(23,51)Latr
C      write(23,52)Depthr
C      write(23,52)Modepthr
C      write(23,111)nyearr
C      write(23,111) nmonthr
C      write(23,111) ndayr
C      write(23,111) nhourr
C      write(23,111) nobr
C      write(23,111)nmsr
C      write(23,111) n
C      do11 k=1,n
C   11 write(23,56) z(k,jjj),T(k,3),s(k,3),O2(k,3)
   56 format(2x,f5.0,1x,f8.3,1x,f8.3,1x,f6.2)
C
      goto555
  112 continue
      close(unit=21)
      close(unit=22)
C      close(unit=23)
      stop '*** E N D ***'
      end
```

Read-22

```
      program BSHR
C V.Guretsky, AWI, Feb 1992
C
C
      real*4 zz(500),
     *       tst(42), sst(42), oxst(42), zst(42),
     *       PST(42),SIST(42),AZOTST(42)
C
      integer*4 crunu,stnum
C*********************************************
C
      open(22,file='intbsh.dat',status='old')
  222 continue
  301 format(2x,f9.4,1x,f9.4)
  302 format(2x,5i7)
  344 format(2x,i3,2x,a4,i4,i4)
501   format (1x,f5.0,6(1x,f10.3))
CWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
      read(22,344,end=333) iseq,a4
      read(22,302) crunu,stnum
      read(22,301) xlon,xlat
      read(22,302) iday,imon,iyear,ihour,imin
      read(22,302) imaxod,nobs,iwmosq,ibot
      read(22,302) mmax
C
      do 11 k=1,mmax
   11 read(22,501) zst(k), tst(k), sst(k), oxst(k),pst(k),sist(k),
     *azotst(k)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      go to 222
  333 continue
      type*,'nstat=',iseq
      close(22)
c
c ISEQ: sequntial number in the current file
c A4: NODC code
c XLON: longitude in decimal degrees
c XLAT: latitude in decimal degrees
c IDAY: day of observation
c IMON: month of observation
c IYEAR: year of observation
c IHOUR: hour of observation
c IMIN: minute of observation
c IBOT: bottom depth
c IMAXOD: maximum observed depth
c NOBS: number of observed depths
c IWMOSQ: ten degree WMO square number
c DEPTH, TEMPERATURE, SALINITY, OXYGENE, PHOSPHATE, SILICATE, NITRATE
C
      stop '***END***'
       end
```

Read - 21

```
      program readawi
C
C  read interpolated data of Polarstern Cruises
C
       character*30 file
       integer*4 Crunu
       REAL*4 Z(42), T(42), S(42)
C
      type*,'file name'
      accept30,file
   30 format(a30)
      open(20,file=file,status='old')
C=======================================
C     input files are in the directory OTH$daten:[socean.awi] :
C     ant2i.dat
C     ant3i.dat
C     ant5i.dat
c     ant51i.dat
C     ant7i.dat
C     ant7i.dat
C=======================================
c
  222 continue
C
      read(20,*,end=333) NSEQ ! seq number in the file
      read(20,*) Crunu ! Cruise_Number
      read(20,*) ISTAT         ! station number
      read(20,*) PHI,AMBDA  ! Latitude, Longitude (grad)
      read(20,*) NDA,MON,NYE,NHO,MIN ! day, Month, Year, Hour, Min
      read(20,*) MBDEPTH, IZLAST ! Bottt_Depth (m) Max_Obs Depth (m)
      read(20,*) NUMOBS , NUMST! Number_Obs_Levels   Num_Stand_Levels
      read(20,*) MSQ ! Marsden square
C
      type*,Nseq
      type*,Crunu,ISTAT
      type*,PHI, AMBDA
      type*,NDA,MON,NYE,NHO,MIN
      type*,MBDEPTH,IZLAST
      type*,NUMOBS,NUMST
      type*,MSQ
      do 9 k=1,NUMST
      read(20,*) KK,Z(k),T(k),S(k)
      type*,KK,Z(k),T(k),S(k)
    9 continue
C====================
      go to 222
  333 continue
      close(20)
      stop
      end
```

Read-20

```
      program READAWI1
C
C/ 1 /  Read files with Polarstern data from G.Rohard's directory:
C                             SCR$DISK1:[rohardt.ctd]
C  Each file of these data corresponds to one hydrographic station
C  Files are kept in subdirectories: ANT2.DIR  ANT3.DIR  ANT5.DIR
C                                   ANT5-1.DIR  ANT7.DIR  ANT8.DIR
C
C  All files there have the same structure of the name,i.g.:
C      NNNNN.DAT,  whre NNNNN is "Polarstern Station_Number".
C  Names of station-files for each cruise are kept in the corresponding
C  files: ANT2NUMBER.DAT   ANT3NUMBER.DAT  ANT5NUMBER.DAT  ANT51NUMBER.DAT
C  ANT7NUMBER.DAT  ANT8NUMBER.DAT
C
C/ 2 /  Merge all  files, corresponding to the same cruise, into
C  a single file. Names of output files are as follows: ANT2.dat
C   ANT3.dat  ANT5.dat  ANT51.dat  ANT7.dat  ANT8.dat
C-------------------------------------------------------------------
C
      character fileroh*12,filegur*50,filename*50
      character*1 char
      real P(5000),T(5000),s(5000),C(5000)
C
      type*,'file of STATION-file-names in G.Rohard's directory'
      accept100, filename
  100 format(A50)
      open(21,file=filename,status='old')
C
      type*,'output file name (new file containing all
     *stations of a cruise)'
      accept100,filegur
      open(23,file=filegur,status='new')
C-------------------------------------------------------------------
      nstat=0
  222 continue
      read(21,101,end=333)fileroh
  101 format(a12)
C
      open(22,file=fileroh,status='old',READONLY)
C
C READ HEADER
      read(22,88) ISTAT
   88 format(1x,i5)
      nstat=nstat+1
      type *, nstat,ISTAT
      read(22,*) NGRADP,AMINP,NGRADL,AMINL
CC      type*,NGRADP,AMINP,NGRADL,AMINL
      read(22,*) NDA,MON,NYE,NHO,MIN
CC      type*,NDA,MON,NYE,NHO,MIN
      read(22,*) MBDEPTH
CC      type*,MBDEPTH
      read(22,*) MLAST
CC      type*,MLAST
      do k=1,5
      read(22,203)CHAR
  203 format(A1)
      if(char.eq.'#') go to 2
      end do
    2 continue
C
C
C READ OBSERVED LEVEL DATA
      do 3 k=1,10000
      read(22,*,end=33)IN,P(k),T(k),C(k),S(k)
CC      type*,IN,P(k),T(k),S(k)
      Kmax=k
    3 continue
   33 continue
C
C-----------------------------------------------------------------------
C UNFORMATTED OUTPUT
C 1 WRITE HEADER
      write(23,*) ISTAT
      write(23,*) NGRADP,AMINP,NGRADL,AMINL
      write(23,*) NDA,MON,NYE,NHO,MIN
      write(23,*) MBDEPTH
      write(23,*) MLAST
      write(23,*) KMAX
C 2 WRITE OBSERVED LEVEL DATA
      do 5 k=1,KMAX
      write(23,*) K,P(k),T(k),S(k)
    5 continue
C
C-----------------------------------------------------------------------
      go to 222
  333 continue
      close(21)
      close(22)
      close(23)
      stop
      end
```

Read - 19

```
      program readargent
c    this program reads Argentine data
c      V.Guretsky, AWI, June 1991
c
      real*4 tem(42), sal(42), oxy(42),z(42)
      character file1*15, file2*15, country*2,ship*2,cruise*3,
c
      open(22,file='interarg4.dat',status='old')
    2 continue
      read(22,202,end=3) nseq,NCRUISE,nstat, ongitud,atitud
      read(22,203) nyear,month,nday,
     *nhour,nmin,depth,modepth,K,msq10
      read(22,204)country
      read(22,204)ship
      read(22,205)cruise
  202 format(2x,3i7,2f8.2)
  203 format(10i7)
  204 format(2x,a2)
  205 format(2x,a3)
c
      do kk=1,K
      read(22,103) z(kk), tem(kk), sal(kk),oxy(kk)
  103 format(2x,f5.0,2f7.3,f6.2)
      end do
      go to 2
    3 continue
      close(unit=22)
      stop '***END***'
      end
```

read 1

reading of Kuropatkin

```
      program read1
C     Guretsky, AWI, 21 June 1990
      real*4 z(42), t(42), s(42), ox(42)
C
      character file1*15
      type*,'name of the input file'
      accept 100, file1
  100 format(a15)
      open(unit=21, file=file1,status='old')
C
  222 continue
      read(21,101,end=333) nseq, nc, ns, ongitud, atitud, nyear, nmo, nda, nho,
     * nde, mod, nz, msq
      type 101,
     * nseq,nc,ns,ongitud,atitud,nyear,nmo,nda,nho,nde,mod,nz,msq
      read(21,101) ni
      type*,ni
      do 1 k=1,ni
      read(21,102) z(k), t(k), s(k), ox(k)
    1       type 102, z(k), t(k), s(k), ox(k)
      go to 222
C
C     nseq - sequential number of station in the file
C     nc - cruise number
C     ns - station_number
C     ongitud - Longitude
C     atitude - Latitude
C     nyear - Year
C     nmo - month
C     nda - day
C     nho - hour
C     nde - Bottom_Depth
C     mod - Max_Obse_Depth
C     nz - number_obse
C     msq - Marsden_Square
C     ni - number of standard (interpolated) levels
C
  333 continue
  101 format(2x,3i7,2f8.2,9i7)
  102 format(2x,f5.0,3f8.3)
      stop
      end
```

Read-19

```
      program gordcr3
C     reads Gordon cruises from HUBER FILE
C     AND CREATE NEW FILE OF THE SAME SIZE WITH CRUISE NUMBERS
C
      character*5 Ship1(100), S
      character*25 ship2(100)
      integer*4 Crunu(100), ID
C
      open(unit=22,file='SHIPGORD2.dat',status='old')
    7 continue
      read(22,11,end=300)k,Ship1(k),Ship2(k),Crunu(k)
      go to 7
  300 continue
   11 format(2x,i3,2x,a5,2x,a25,2x,i7)
C
      open(unit=20,file='headers.fil',status='old')
      open(unit=24,file='shipgord3.dat',status='new')
      N=0
      ID=100000
      NSEQ=0
C
C  READ FILE "HEADERS.FIL" (ONLY FIRST VALUE AS CHARACTER*5)
C
  100 read(20,10,end=200)S
      Nseq=Nseq+1
      ID=ID+1
   10 format(1x,A5)
C
      jj=0
C
C  FIND SHIPCODE IN "SHIPGORD2.DAT" WHICH EQUAL TO SHIPCODE  S  IN
C                                        "HEADERS.FIL"
      do 8 j=1,k
      if(S.EQ.Ship1(j))jj=j
    8 continue
      if(jj.eq.0) go to 9
      write(24,50)Nseq,ID,S,Ship2(jj),Crunu(jj)
   50 format(2x,i4,2x,i7,2x,a5,2x,a25,2x,i7)
      go to 100
C
C
C
  200 continue
      go to 19
    9 continue
      type*,'NO SUCH CRUISE  N=',N, S
   19 continue
      close(24)
      close(22)
      close(20)
      stop
      end
```

Read-18

```
       program gordcr41
C
C  INSERT CRUISE NUMBERS FOR THE GORDON PART OF Station table
C  ACCORDING TO THE FILE SHIPGORDNEW.DAT obtained by the program
C  GORDCR31
       EXTERNAL err_handler
       External msg_handler
       include '(fsybdb) '
       Integer*4  numer, dbproc, login,return_code,error
C
       character*5 Ship1, S
       character*25 ship2
       integer*4 Crunu, ID
C
       call fdberrhandle(err_handler)
       call fdbmsghandle(msg_handler)
       login=fdblogin()
       call fdbsetluser(login,'SOCEAN')
       call fdbsetlpwd(login,'Victor')
       dbproc=fdbopen(login,NULL)
       call fdbuse(dbproc,'SouthernOceanDB')
       open(unit=24,file='SHIPGORDNEW.dat',status='old')
     8 continue
       read(24,50,end=200)Nseq,ID,Ship1,Ship2,Crunu
    50 format(2x,i4,2x,i7,2x,a5,2x,a25,2x,i7)
C
       call fdbfcmd(dbproc,'Execute Gordcr4 %d,%d',ID,Crunu)
       call fdbsqlexec(dbproc)
       TYPE*,ID
       go to 8
   200 continue
       call fdbexit()
       close(24)
       stop
       end
C      Error und Message Handler fuer
C      embedded SQL-Programme. In diesen mit
C      INCLUDE '(ERRMSG)' includen.
C
C      Error Handler
C      -------------
C      ERR_HANDLER - This funtion may be coded within the same program
C      or as a separate file that is compiled/linked.
C
       INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
       include '(fsybdb)'
C
C       EXTERNAL        err_handler
C       EXTERNAL        msg_handler
C
       INTEGER*4       dbproc
       INTEGER*4       severity
       INTEGER*4       errno
       INTEGER*4       oserrno
       INTEGER*4       length
       INTEGER*4       return_code
C
       CHARACTER*(80)  message
C
          length = fdberrstr(errno,message)
          type *, 'DB-LIBRARY error: ', message
C
C      Check for operating system errors
C
          length = 0
```

Road-17

```
          message = ' '
          length = fdboserrstr(oserrno, message)
C
          if (oserrno .ne. DBNOERR) then
             type *, 'Operating-system error: ', message
          end if
C
          return_code = fdbdead(dbproc)
C
          if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2       (severity .eq. EXSERVER)) then
             err_handler = INT_EXIT
C
          else
             err_handler = INT_CANCEL
          end if
C
          END
C
C       Message Handler
C       ----------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2             msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4       dbproc
      INTEGER*4       msgno
      INTEGER*4       msgstate
      INTEGER*4       severity
C
      CHARACTER*80    msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
      program gordcr4
C     reads Gordon cruises from HUBER FILE
C     AND CREATE NEW FILE OF THE SAME SIZE WITH CRUISE NUMBERS
C
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
      Integer*4  numer, dbproc, login,return_code,error
C
      character*5 Ship1, S
      character*25 ship2
      integer*4 Crunu, ID
C
      call fdberrhandle(err_handler)
      call fdbmsghandle(msg_handler)
      login=fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login,'Victor')
      dbproc=fdbopen(login,NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
      open(unit=24,file='SHIPGORD3.dat',status='old')
    8 continue
      read(24,50,end=200)Nseq,ID,Ship1,Ship2,Crunu
   50 format(2x,i4,2x,i7,2x,a5,2x,a25,2x,i7)
C
      call fdbfcmd(dbproc,'Execute Gordcr4 %d,%d',ID,Crunu)
      call fdbsqlexec(dbproc)
      TYPE*,ID
      go to 8
  200 continue
      call fdbexit()
      close(24)
      stop
      end

C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
      include '(fsybdb)'
C
C      EXTERNAL        err_handler
C      EXTERNAL        msg_handler
C
      INTEGER*4       dbproc
      INTEGER*4       severity
      INTEGER*4       errno
      INTEGER*4       oserrno
      INTEGER*4       length
      INTEGER*4       return_code
C
      CHARACTER*(80)  message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
```

Real 16

```
          message = ' '
          length = fdboserrstr(oserrno, message)
C
          if (oserrno .ne. DBNOERR) then
             type *, 'Operating-system error: ', message
          end if
C
          return_code = fdbdead(dbproc)
C
          if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2       (severity .eq. EXSERVER)) then
             err_handler = INT_EXIT
C
          else
             err_handler = INT_CANCEL
          end if
C
          END
C
C       Message Handler
C       ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2            msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4       dbproc
      INTEGER*4       msgno
      INTEGER*4       msgstate
      INTEGER*4       severity
C
      CHARACTER*80    msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
      program gordcr31
C     reads Gordon cruises from HUBER FILE
C     AND CREATE NEW FILE OF THE SAME SIZE WITH CRUISE NUMBERS
C
      character*5 Ship1(100), S
      character*25 ship2(100)
      integer*4 Crunu(100), ID
C
      open(unit=22,file='SHIPGORD2.dat',status='old')
    7 continue
      read(22,11,end=300)k,Ship1(k),Ship2(k),Crunu(k)
      type*,k,Ship1(k),Ship2(k),Crunu(k)
      go to 7
  300 continue
   11 format(2x,i3,2x,a5,2x,a25,2x,i7)
C
      open(unit=20,file='headersnew.fil',status='old')
      open(unit=24,file='shipgordnew.dat',status='new')
      N=0
      ID=100000
      NSEQ=0
C
C    BEGIN TO READ HEADERS1.fil (ONLY THE FIRST VALUE AS CHARACTER*5)
C
      do 100 iii=1,6313
      read(20,12,end=200)S
      type*,iii,S
      ID=ID+1
   12 format(1x,a5)
C
C  FIND WICH SHIPCODE FROM "SHIPGORD2.DAT" EQUAL TO SHIPCODE  S  FROM
C                                                   "HEADERS1.FIL"
      jj=0
      do 8 j=1,k
      if(S.EQ.Ship1(j))jj=j
    8 continue
      nseq=iii
      if(jj.eq.0) go to 9
      write(24,50)iii,ID,S,Ship2(jj),Crunu(jj)
   50 format(2x,i4,2x,i7,2x,a5,2x,a25,2x,i7)
  100 continue
C
C
  200 continue
      go to 19
    9 continue
      type*,'NO SUCH CRUISE  N=',Nseq, S
   19 continue
      close(24)
      close(22)
      close(20)
      stop
      end
```

Read 15

Readmuin
June 91

```
      program readmuin
Cread interpolated MUENCH DATA
C
C     V.Guretsky, AWI, JUNE 1991
C
      character file1*15, file2*15
C
      integer*4 NCRU
C
      real*4    zg1(5000),tg1(5000),sg1(5000),zst(42),
     *      fob1(5000), zob1(5000) ,TST(42),SST(42)
C
      data zst /0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,
     * 250.,300.,350.,400.,500.,600.,700.,750.,800.,900.,
     * 1000.,1100.,1200.,1300.,1400.,1500.,1750.,2000.,2250.,2500.,
     * 2750.,3000.,3250.,3500.,3750.,4000.,4500.,5000.,5500.,6000.,
     * 6500.,7000./
C     ---------------------------------------
  100 format(a15)
C
       type*, 'Name of input file'
       accept 100,file1
       open(unit=20, file=file1,status='old')
C
C---------------------------------------------------
  222 continue
      read(20,202,end=333) nseq,NCRU,numst, ongitud,atitud
      read(20,203) nyear,nmonth,nday,
     *nhour,nmin,ndepth,modepth,nlev,msq
  104 format(5(1x,f7.2,2f7.3))
  202 format(2x,3i7,2f8.2)
  203 format(10i7)
   22 format(2x,i3,2x,f6.1,2f7.3)
      read(20,22) J
C
      do9 i=1,J
      read(20,22) ii, zg1(i),Tg1(i),Sg1(i)
    9 continue
C================================================================
      go to 222
  333 continue
C
      type*,'total number of stations in the file is ',nseq
      close(unit=20)
      stop '********* E N D *********'
      END
```

Read-14

Mudbarmeter
June 91

```
      program mudbarmeter
C     converts P(dbar) into Z(meters) for the levels of
C     observations and for the max_obs_depth in the header
C
C     Guretsky, AWI, June 1991
C
      real p(9000), t(9000), s(9000), o2(9000),PMAX,z(9000)
      character*20 Cruise
      character file1*15, file2*15
C
   50 format(a2)
C
      type*,'name of the input file'
      accept 100, file1
  100 format(a15)
      open(unit=21, file=file1
     *,status='old')
      type*,'name of the output '
      accept 100, file2
      open(unit=22, file=file2
     *,status='new')
C
  900 continue

C
C_____________________________INPUT_____________________________
      read(21,202,end=901) nseq,NCRU,numst, ongitud,atitud
      read(21,203) nyear,nmonth,nday,
     *nhour,nmin,ndepth,modepth,nlev,msq
      read(21,103) (p(k), t(k), s(k),k=1,nlev)
  202 format(2x,3i7,2f8.2)
  203 format(10i7)
  103 format(5(1x,f5.0,2f7.3))
C    _______________________________________________
C
      do i=1,nlev
      call condbar(p(i),PH,z(i))
      end do
C
      modepth = z(nlev)
C
C                 OUTPUT:
C    _______________________________________________
      write(22,202) nseq,NCRU,numst, ongitud,atitud
      write(22,203) nyear,nmonth,nday,
     *nhour,nmin,ndepth,modepth,nlev,msq
      write(22,104) (z(k), t(k), s(k),k=1,nlev)
  104 format(5(1x,f7.2,2f7.3))
C-------------------------------------------------------
      go to 900
  901 continue
      close(22)
      close(21)
      stop
      end
```

Read-13

Reading

```
      program reading
c          this program read the interpolated data from the disk
       integer*2 a(12),t(42),s(42),ox(42),z(42)
      open(12,file='oth$daten:[vgurets]disk2.dat',
     *status='old',access='sequential',
     *recl=276,form='formatted',recordtype='fixed')
c
      type *,' how many stations would you like to read ? '
      accept*, nst
        do 33 n = 1, nst
      read(12,100,end=3)a,t,s,ox
      m = n
  100 format(138a2)
c              these are the standard levels depths:
         data z / 0, 10, 20, 30, 50, 75, 100, 125, 150, 200, 250,
     *            300, 350, 400, 500, 600, 700, 750, 800, 900,
     *            1000, 1100, 1200, 1300, 1400, 1500, 1750, 2000,
     *            2250, 2500, 2750, 3000, 3250, 3500, 3750, 4000,
     *            4500, 5000, 5500, 6000, 6500, 7000 /
c
c        a(1) - archiv number of cruise
c        a(2) - cruise number of station
c        a(3) - latitude (in degrees * 100)
c        a(4) - longitude (in degrees * 100)
c        a(5) - year
c        a(6) - month
c        a(7) - day
c        a(8) - hour
c        a(10) - depth of the deepest observed level
c        a(11) - total number of observed levels
c        a(12) - Marsden square
c
c        t - array of interpolated temperature values ( * 1000 )
c        s - array of interpolated salinity values ( ( S - 30 ) * 1000 )
c        ox - array of interpolated oxygen values ( * 100 )
c
      type 101,a
      do k=1,42
      type 102, z(k),t(k),s(k),ox(k)
        end do
c
   33 continue
      goto4
    3 continue
      type*,'end of file'
      type*,' there are ', i6, ' stations in the file'
    4 close( 12)
  101 format(1x,12i6)
  102 format(1x,4(i7))
       end
```

Read-12

readgocr

```
      program readgocr
C     reads Gordon cruises
C
      character*5 Ship(7000),S
      open(unit=20,file='headers.fil',status='old')
      N=0
      L=0
  100 read(20,10,end=200)S
      N=N+1
      if(N.gt.1) go to 2
      Ship(1)=S
    2 continue
      M=0
      do3 k=1,L
    3 if(Ship(k).eq.S)M=1
      if(M)4,5,4
    5 L=L+1
      Ship(L)=S
    4 continue
   10 format(1x,A5)
      go to 100
  200 continue
      open(unit=22,file='SHIPGORD.dat',status='new')
      do 7 k=1,L
      write(22,11)k,Ship(k)
    7 type 11,k,Ship(k)
   11 format(2x,i4,2x,A5)
      close(22)
      close(20)
      stop
      end
```

Read-11

Read france
18.12.96

[illegible] [illegible]

```
      program readfrance
C
C     V.Guretsky, AWI, 13 DECEMBER 1990
C
C     READ FILE OTH$DATEN:[socean.gonella]GONELLA7.dat
C
C
C     THESE ARE DATA provided by National Museum of Natural History
C     in Paris
C      THERE ARE 277 Oceanographic stations, obtained during 7 cruises
C      of research vessel   "Marion Dufresne"
C
C     THESE Cruises will have Cruise_Numbers between 57001 and 57007
C
C                                                        new cruise-numbers
C
      real*4   zst(42),
     *        TST(42),SST(42),OST(42)
C
      integer*4 numer
C
C     ------------------------------------
       open(unit=22, file='oth$daten:[socean.gonella]gonella7.dat'
     *,status='old')
C-----------------------------------------------------------------
  222 continue
      read(22,256,end=333) Numer, nstat, ALA, PHI, ndepth,MOD,
     * nyear, month,
     *nday, NTIME,NZ
C
      type 256,Numer,nstat,ALA,PHI,ndepth,MOD,nyear,month,nday,
     *ntime,nz
C
      read(22,22)mmax
      do i=1,mmax
      read(22,22) j, zst(i),TST(i),SST(i),OST(i)
      end do
      go to 222
  333 continue
C
C       VARIABLE DESCRIPTION
C  NUMER - Cruise Number                       Marsden. Square is missing
C  nstat - Station number within the Cruise
C  ALA - Longitude
C  PHI - Latitude
C  ndepth - Bottom depth
C  mod - max_obse_depth
C  nyear - Year
C  month - month
C  nday - day
C  ntime - Hour
C  nz - NUMBER of observed levels
C  mmax - number of interpolated levels
C  zst - Standard depth array
C  tst - temperature at the standard levels
C  sst - salinity at the Standard levels
C  ost - oxygen at the standard levels
C
C
   22 format(2x,i3,2x,f6.1,f7.3,f7.3,f6.2)
  256 format(2x,i6,1x,i4,1x,f9.4,1x,f9.4,1x,i4,1x,i4,1x,i4,1x,
     *i2,1x,
     *i2,1x,
     *i2,1x,i3)
      close(unit=22)
      close(unit=20)
```

Read-10

Kurop. For
August 90

```
      program KUROP
C    this program converts file KUROP1.DAT in t the form
C    suitable for the Data_Set
C      V.Guretsky, AWI, August, 1990
C
      real*4 tem(80), sal(80), oxy(80),z(80)
C
      character file1*15, file2*15
C
      KOUNT=0
      type*,'input file name'
      accept 100, file1
  100 format(a15)
      type*,'name of the output file'
      accept 100, file2
      open(unit=21,file=file1,status='old')
      open(unit=22,file=file2,status='new')
  222 continue
      read(21,111,end=333)i8
  111 format(i4)
      read(21,101) nyear,month,nday,nhour,min,lat,lon,ndepth,nseq
  101 format(i2,1x,i2,1x,i2,1x,i2,1x,i2,1x,i4,1x,i5,1x,i4,1x,i3)
      type*,nseq,nyear,month,nday,lat,lon
      i=1
    8 continue
      read(21,102,end=91)z(i), Tem(i), Sal(i)
      if(z(i).eq.8888.)backspace(21)
      if(z(i).eq.8888.)go to 9
  102 format(f4.0,1x,f4.2,1x,f5.3)
      i=i+1
      go to 8
    9 continue
      go to 92
   91 continue
      ind=1
   92 continue
      n=i-1
      type*,'N=',n
C                         CONVERSION OF TYPES FOR COORDINATES
C
      p=float(lat)
      r=p/100.
      s=aint(r)
      t=r-s
      u=t*5./3.
      atitud=s+u
C
      p=float(lon)
      r=p/100.
      s=aint(r)
      t=r-s
      u=t*5./3.
      ongitud=s+u
C
      NCRU=-23011
      NYEAR=1900+NYEAR
      MODEPTH=IIFIX(z(n))
      MSQ=999
      KOUNT=KOUNT+1
C_______________________________OUTPUT___________________________
      write(22,202) nseq,NCRU,nseq, ongitud,atitud,nyear,month,nday,
     *nhour,ndepth,modepth,n,msq
  202 format(2x,3i7,2f8.2,9i7)
      do 2 k=1,n
      write(22,103) z(k), tem(k), sal(k), oxy(k)
  103 format(2x,f5.0,3f8.3)
```

Real-9

READAARI.for
reads AARI Data
5.3.91

```
      program readaari
C     Guretsky, AWI, 5 March 1991
      real*4 z(42), t(42), s(42), ox(42)
C
      character file1*15
      type*,'name of the input file'
      accept 100, file1
  100 format(a15)
      open(unit=21, file=file1,status='old')
C
  222 continue
      read(21,101,end=333) nseq, nc, ns, ongitud, atitud, nyear,
     * nmo, nda, nho,
     * nde, mod, nz, msq
  101 format(2x,3i7,2f8.2,9i7)
      type 101,
     * nseq,nc,ns,ongitud,atitud,nyear,nmo,nda,nho,nde,mod,nz,msq
      read(21,101) ni
      type*,ni
      do 1 k=1,ni
      read(21,102) z(k), t(k), s(k), ox(k)
    1       type 102, z(k), t(k), s(k), ox(k)
      go to 222
C
C     nseq - sequential number of station in the file
C     nc - cruise number
C     ns - station_number
C     ongitud - Longitude
C     atitude - Latitude
C     nyear - Year
C     nmo - month
C     nda - day
C     nho - hour
C     nde - Bottom_Depth
C     mod - Max_Obse_Depth
C     nz - number_obse
C     msq - Marsden_Square
C     ni - number of standard (interpolated) levels
C
  333 continue
  102 format(2x,f5.0,3f8.3)
      stop
      end
```

Read-4

OTHSDATEN : [SOCEAN.FOR] READHEINZ.FOR

Standard_Depth_Level

initial_Data are

Readheinz

11.90

```
      program readheinz
C
C     READ THE DATA PROVIDED BY W.HAINES FROM Lamont-Doherty
C      Geological Observatory
C
C     There are 617 stations
C
C     V.Guretsky, AWI, November 1990
C
      real*4   zst(42)
     *         ,TST(42),SST(42),OST(42)
C
      integer*4 CRUNU
C
C
C     nseq - sequential number
C     Crunu - Cruise_Number
C     ns - station_number
C     ongitud - Longitude
C     atitude - Latitude
C     nyear - Year
C     nmo - month
C     nda - day
C     nho - hour
C     nde - Bottom_Depth
C     mod - Max_Obse_pressure bzw. _depth
C     nz - number of observed levels
C     mmax - number of interpolated values which are to be kept in the
C            SO_Data_Base
C
       open(unit=21, file='oth$daten:[socean.heinz]heinzint1.dat',
     * status='old')
  222 continue
  102 format(2x,f7.2,1x,3f8.3)
      read(21,401,end=333) nseq, CRUNU, ns, ongitud, atitud, nyear, nmo, nda,
     * nho,
     * nde, mod, nz, msq
      read(21,401)mmax
      do 11 k=1,mmax
   11 read(21,102) zst(k), tst(k), sst(k), ost(k)
  401 format(2x,3I7,2x,2f9.4,2x,8i5)
      M=M+1
      type*,M
      go to 222
C
  333 continue
      close(unit=21)
      close(22)
      stop '********* E N D *********'
      END
```

Read-7

Readnowl
Oktober 90

```
      program readnowl
C   this program converts file S_OCEAN.DAT from W.NOWLIN
C    in to the form
C   suitable for the Data_Set
C     V.Guretsky, AWI, OCTOBER, 1990
C
      real*4 tem(80), sal(80), oxy(80),z(80)
C
      character file1*15, file2*15,SHIP*2,anumst*4
C
      type*,'input file name'
      accept 100, file1
  100 format(a15)
      type*,'name of the output file'
      accept 100, file2
      open(unit=21,file=file1,status='old')
      open(unit=22,file=file2,status='new')
      kount=0
  222 continue
      read(21,111,end=333,err=334)ship,ncode,numst,alat,alon,nhour,nday,
     *nmonth,nyear,numobs,ndepth
      go to 335
  334 continue
      backspace(21)
      read(21,113,end=333)ship,ncode,anumst,alat,alon,nhour,nday,
     *nmonth,nyear,numobs,ndepth
  113 format(1x,a2,3x,i2,1x,A4,1x,f7.3,1x,f8.3,1x,i4,1x,i2,1x,i2,1x,
     *i2,1x,i2,2x,i4,2x,i4,2x,i4)
      numst=-99
  335 continue
      kount=kount+1
      nhour=nhour/100
  111 format(1x,a2,3x,i2,1x,i4,1x,f7.3,1x,f8.3,1x,i4,1x,i2,1x,i2,1x,
     *i2,1x,i2,2x,i4,2x,i4,2x,i4)
      type *,kount
      type111,SHIP,NCODE,numst,alat,alon,nhour,nday,nmonth,nyear,numobs
     *,ndepth
      i=1
      do 8 k=1,numobs
      read(21,102)z(k), Tem(k),Tpot, Sal(k),Oxy(k)
      if(kount.gt.20) go to 8
      type 102,z(k),Tem(k),Sal(k),Oxy(k)
  102 format(2x,f5.0,2x,f6.3,4x,f6.3,3x,f6.3,4x,f5.2)
    8 continue
      NYEAR=1900+NYEAR
      MODEPTH=IIFIX(z(n))
      MSQ=999
C______________________________OUTPUT______________________________
C      write(22,202) nseq,NCRU,nseq, ongitud,atitud,nyear,month,nday,
C     *nhour,ndepth,modepth,n,msq
  202 format(2x,3i7,2f8.2,9i7)
C      do 2 k=1,n
C      write(22,103) z(k), tem(k), sal(k), oxy(k)
  103 format(2x,f5.0,3f8.3)
C     ____________________________________________________________
    2 continue
      go to 222
  333 continue
      close(unit=21)
      close(unit=22)
      type*,'number of stations=',nseq
      stop '***END***'
      end
```

Read-6

Readnowlin
Oktober 90

```
      program readnowlin
C     READS AND TYPES THE INTERPOLATED DATA PROVIDED BY W.NOWLIN
C     input file:  OTH$DATEN:[socean]NOWLINT.DAT
C V.Guretsky, AWI, October, 1990
C
      real*4 z(80),tem(80), sal(80), oxy(80), ongitud, atitud,
     *       tst(42), sst(42), oxst(42), zst(42), fob1(80), zob1(80)
C
      character file1*15, file2*15
C
      data zst/0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,250.,
     *300.,350.,400.,500.,600.,700.,750.,800.,900.,1000.,1100.,
     *1200.,1300.,
     *1400.,1500.,1750.,2000.,2250.,2500.,2750.,3000.,3250.,3500.,
     *3750.,4000.,4500.,5000.,5500.,6000.,6500.,7000./
C
  100 format(a15)
      open(unit=22, file='NOWLINT.DAT', status='old')
  222 continue
c
  102 format(2x,3i7,2f8.2,9i6)
  103 format(2x,f5.0,3f8.3)
      read(22,102,end=333) nseq, nc, ns, ongitud, atitud, nye, nmo,nda,nho,
     *nde, mod, nz, msq
      type 102, nseq, nc, ns, ongitud, atitud, nye, nmo,nda,nho,
     *nde, mod, nz, msq
C
      read(22,102) mmax
      do 11 k=1, mmax
      read(22,103) zst(k), tst(k), sst(k), oxst(k)
C
   11 type 103,zst(k), tst(k), sst(k), oxst(k)
C
      go to 222
  333 continue
      close(unit=22)
      stop '***END***'
       end
```

Read-5

Readjap

```
      program READJAP
C
C***** V.Guretsky, AWI, February 1991*****
C
C    READ DATA OBTAINED FROM TOKYO UNIVERSITY OF FISHERIES
C
C    Total number of stations within this data set is 188
C
C    Data obtained by R/V "Umitaka-Maru"
C
      real*4
     *        tst(42), sst(42), oxst(42), zst(42)
C
      integer*2 crunu
C
C-----------------------------------------------------------
C     VARIABLES DESCRIPTION
C
C      mseq - sequential number in the file
C      CRUNU - cruise number
C      numstat -  station number in the cruise
C      A - Longitude
C      B - latitude
C      nyear - year
C      month - month
C      nday - day
C      nhour , minut - Time of observation
C      ndep - bottom depth
C      modepth - maximum observed depth
C      n  - number of observed levels
C      msq - marsden square (absent)
C      mmax - number of interpolated levels
C
C
      open(unit=22, file='OTH$DATEN:[socean.JAPAN]TOKYOINT.DAT',
     * status='old')
      do NNNN=1,188
      read(22,202) mseq,CRUNU,numstat,A,P,nyear,month,nday,
     *nhour,minut,ndep,modepth,n,msq
      type 202, mseq,CRUNU,numstat,A,P,nyear,month,nday,
     *nhour,minut,ndep,modepth,n,msq
      read(22,102) mmax
      do 11 k=1, mmax
      read(22,103) zst(k), tst(k), sst(k), oxst(k)
   11 type 103, zst(k), tst(k), sst(k), oxst(k)
      end do
C
  103 format(2x,f5.0,3f8.3)
  102 format(2x,i3)
  202 format(2x,3i7,2f8.2,9i7)
C
      close(unit=22)
      stop '***END***'
       end
```

Read-4

Readjap 2

```
      program readjap2
C    this program converts file JAPAN5.dat into the form
C    suitable for the Data_Set
C      V.Guretsky, AWI, Febr 1991
       real*4 tem(80), sal(80), oxy(80),z(80)
C
       character file1*15, file2*15
       character*1 BL,E(3)
       integer*2 Crunu
       BL=' '
       type*,'Cruise_Number'
       accept 110,Crunu
   110 format(i6)
C
       KOUNT=0
       type*,'input file name'
       accept 100, file1
   100 format(a15)
       type*,'name of the output file'
       accept 100, file2
       open(unit=21,file=file1,status='old')
       open(unit=22,file=file2,status='new')
       kount=0
   222 continue
       read(21,101,end=92) nseq,nstat,PG,PM,AG,AM,nday,month,nyear,
      *nhour,minut,
      *ndep
   101 format(i2,i4,f3.0,f4.1,f5.0,f4.1,3i3,i3,i2,i5)
       type*,nseq,nstat,PG,PM,AG,AM,nday,month,nyear,nhour,minut,ndep
       i=1
     8 continue
       if(i.eq.1) go to 147
       read(21,145)E
   145 format(3a1)
       if(E(3).eq.BL)go to 91
       if(E(3).ne.BL.and.i.ne.1) backspace(21)
       if(E(1).eq.'-')go to 95
   147 continue
       read(21,102,end=91)z(i), Tem(i), Sal(i),Oxy(i)
       type*,z(i), Tem(i),Sal(i),Oxy(i)
       i=i+1
       go to 8
    91 continue
       backspace(21)
       go to 96
    95 ind=99
    96 continue
       i=i-1
   102 format(f4.0,1x,f5.3,1x,f5.3,1x,f3.2)
C
       n=i
       type*,'n=',n
C
       PMM=PM*100./60.
       amm=am*100/60.
       P=PG+PMM*0.01
       P=-1.*P ! GET LATITUDE
C
       if(AG.GE.0.)A=AG+AMM*0.01
       if(AG.lt.0.)A=AG-AMM*0.01 ! GET LONGITUDE
C
       NYEAR=1900+NYEAR
       MODEPTH=IIFIX(z(n))
       MSQ=999
       KOUNT=KOUNT+1
C_________________________________OUTPUT______________________________
```

Read-3

```
      write(22,202) nseq,CRUNU,nstat,A,P,nyear,month,nday,
     *nhour,minut,ndep,modepth,n,msq
  202 format(2x,3i7,2f8.2,9i7)
      do 2 k=1,n
    2 write(22,103) z(k), tem(k), sal(k), oxy(k)
  103 format(2x,f5.0,3f8.3)
C    __________________________________________________
      if(ind.eq.99) go to 92
      go to 222
   92 continue
      type*,'KOUNT=',kount
      close(unit=21)
      close(unit=22)
      stop '***END***'
      end
```

Readjap1

Febr. 91

```
      program readjapl
C   this program converts file JAPANUMIT.dat into the form
C   suitable for the Data_Set
C     V.Guretsky, AWI, Febr 1991
      real*4 tem(80), sal(80), oxy(80),z(80)
C
      character file1*15, file2*15
      character*1 BL,E(3)
      integer*2 Crunu
      BL=' '
      type*,'Cruise_Number'
      accept 110,Crunu
  110 format(i6)
C
      KOUNT=0
      type*,'input file name'
      accept 100, file1
  100 format(a15)
      type*,'name of the output file'
      accept 100, file2
      open(unit=21,file=file1,status='old')
      open(unit=22,file=file2,status='new')
      kount=0
  222 continue
      read(21,101,end=92) nseq,nstat,PG,PM,AG,AM,nday,month,nyear,
     *nhour,minut,
     *ndep
  101 format(i2,i4,f3.0,f4.1,f5.0,f4.1,3i3,i3,i2,i5)
      type*,nseq,nstat,PG,PM,AG,AM,nday,month,nyear,nhour,minut,ndep
      i=1
    8 continue
      if(i.eq.1) go to 147
      read(21,145)E
  145 format(3a1)
      if(E(3).eq.BL)go to 91
      if(E(3).ne.BL.and.i.ne.1) backspace(21)
      if(E(1).eq.'-')go to 95
  147 continue
      read(21,102,end=91)z(i), Tem(i), Sal(i),Oxy(i)
      type*,z(i), Tem(i),Sal(i),Oxy(i)
      i=i+1
      go to 8
   91 continue
      backspace(21)
      go to 96
   95 ind=99
   96 continue
      i=i-1
  102 format(f4.0,1x,f4.2,1x,f5.3,1x,f3.2)
C
      n=i
      type*,'n=',n
C
      PMM=PM*100./60.
      amm=am*100/60.
      P=PG+PMM*0.01
      P=-1.*P ! GET LATITUDE
C
      if(AG.GE.0.)A=AG+AMM*0.01
      if(AG.lt.0.)A=AG-AMM*0.01 ! GET LONGITUDE
C
      NYEAR=1900+NYEAR
      MODEPTH=IIFIX(z(n))
      MSQ=999
      KOUNT=KOUNT+1
C________________________________OUTPUT____________________________
```

Read-2

READJARE. FO
15.4.91

```
      program READJARE
C V.Guretsky, AWI, 15 APRIL 1991
C
      integer*4 crunu, numstat
      real*4 tem(42), sal(42), oxy(42), po(42), si(42), n3(42), zz(42)
C
      open(unit=21,file='oth$daten:[socean.jare]jareall.dat'
     *,status='old')
C         I N P U T
      do 333 L=1,119
      read(21,202) nseq,CRUNU,numstat,A,P,nyear,month,nday,
     *nhour,minut,ndep,modepth,n,msq
C
      type 202, nseq,CRUNU,numstat,A,P,nyear,month,nday,
     *nhour,minut,ndep,modepth,n,msq
C
      read(21,102) mmax
      type 102, mmax
  102 format(2x,i3)
C
      do 2 k=1,mmax
      read(21,103) zz(k),tem(k),sal(k),oxy(k),PO(k),n3(k),SI(k)
    2 type 103,  zz(k), tem(k), sal(k), oxy(k),PO(k),N3(k),SI(k)
C
C
C      VARIABLES:
C NSEQ - sequential number of station in the file
C CRUNU - Cruise Number
C NUMSTAT - Station Number
C A - Longitude
C P - Latitude
C NYEAR - Year
C MONTH - month
C NDAY - Day
C NHOUR - Hour
C MINUT Minutes
C NDEP - BNottom Depth
C MODEPH - Max_Obse_Depth
C N - Number_Obse
C MMAX-Number of interpolated levels
C ZZ - Depth in meters
C TEM - temperature
C SAL - salinity
C OXY - Oxygen
C PO - Phosphatus
C N3 - Nitrate
C SI - Silicate
C
  103 format(2x,f5.0,6f8.3)
  202 format(2x,3i7,2f8.2,9i7)
C ________________________________________________
  333 continue
      close (unit=21)
      stop '***END***'
       end
```

Read-1

Inter-12

Muinter

June 91

```
      program intbsh
C V.Guretsky, AWI, Feb 1992
C
      real xobs, Yobs
C
        character statnr*9,country*2,plat*2,cdir*1
C
C
      real*4 zz(500),tem(500), sal(500), oxy(500), ongi
     *       tst(42), sst(42), oxst(42), zst(42), fobl(500), zobl(500),
     *       PO4(500),SI(500),NO3(500),PST(42),SIST(42),AZOTST(42)
C
      character file1*15, file2*15
C
      integer*2 idepth(500)
      integer*4 NCRUISE,crunu,stnum
C
      data zst/0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,250.,
     *300.,350.,400.,500.,600.,700.,750.,800.,900.,1000.,1100.,
     *1200.,1300.,
     *1400.,1500.,1750.,2000.,2250.,2500.,2750.,3000.,3250.,3500.,
     *3750.,4000.,4500.,5000.,5500.,6000.,6500.,7000./
C
      mseq=0
C*********************************************
C
      open(22,file='bsh9.dat',status='old')
      open(23,file='bshint.dat',status='new')
  190 format(a70)
  222 continue
C



  300 format(2x,i3,2x,a4,1x,a2,1x,a2,1x,a9,1x,a1)
  301 format(2x,f9.4,1x,f9.4)
  302 format(2x,5i7)
  345 format(2x,i3,2x,18a1)
  344 format(2x,i3,2x,a4,i4,i4)
  400   format (1x,i5,6(1x,f10.3))
  347 format(2x,i3,8a1,2x,9a1,2x,2x,a1,1x)
      read(22,344) iseq,a4
      read(22,302) crunu,stnum
      read(22,301) xlon,xlat
      read(22,302) iday,imon,iyear,ihour,imin
      read(22,302)imaxod,nobs,iwmosq
        do 89 i=1,nobs
        read(22,400) idepth(i),tem(i),sal(i),oxy(i),po4(i),
     *  si(i),
     *  no3(i)
       zz(i)=float(idepth(i))
89      continue
  401 format(2x,18a1)
  399 format(2x,i3)
C
200     format(1x,i4,1x,a4,1x,a9,1x,2a2)
201     format(1x,2(1x,f9.4),1x,3i2,1x,i2,':',i2,2(1x,i5),1x,
     *          i3,1x,a3,1x,a1,1x,i4)
C ____________________________________________________
      do 7 k=1,42
      sst(k)=-99.
      oxst(k)=-99.
      tst(k)=-99.
      pst(k)=-99.
      sist(k)=-99.
```

```
        azotst(k)=-99.
     7 continue
C
        nz=NOBS
C
        fmin=-2.3
        fmax=29.
        mt=inter(nz,zz,tem,fmin,fmax,tst,zst,nob2,fob1,zob1)
        fmin=20.
        fmax=36.5
        ms=inter(nz,zz,sal,fmin,fmax,sst,zst,nob2,fob1,zob1)
        fmin=1.
        fmax=1000.
        mox=inter(nz,zz,oxy,fmin,fmax,oxst,zst,nob2,fob1,zob1)
C
        fmin=0.
        fmax=1000.
        mp=inter(nz,zz,PO4,fmin,fmax,pst,zst,nob2,fob1,zob1)
C
        fmin=0.
        fmax=1000.
        msi=inter(nz,zz,si,fmin,fmax,sist,zst,nob2,fob1,zob1)
        fmin=1.
        fmax=1000.
C
        maz=inter(nz,zz,NO3,fmin,fmax,azotst,zst,nob2,fob1,zob1)
C
        mmax=mt
        if(ms.gt.mt)mmax=ms
CWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW
        write(23,344) iseq,a4
        write(23,302) crunu,stnum
        write(23,301) xlon,xlat
        write(23,302) iday,imon,iyear,ihour,imin
        write(23,302) imaxod,nobs,iwmosq
        write(23,302) mmax
C
        if(tst(1).lt.-9..and.sst(1).lt.-9..and.zz(1).lt.4.)go to 14
        go to 15
    14 tst(1)=tem(1)
        sst(1)=sal(1)
        oxst(1)=oxy(1)
        pst(1)=po4(1)
        sist(1)=si(1)
        azotst(1)=no3(1)
    15 continue
C
        do 11 k=1, mmax
        if(oxst(k).lt.1.)oxst(k)=-99.
        if(pst(k).lt.1.)pst(k)=-99.
        if(sist(k).lt.1.)sist(k)=-99.
        if(azotst(k).lt.1.)azotst(k)=-99.
    11 write(23,501) zst(k), tst(k), sst(k), oxst(k),pst(k),sist(k),
      *azotst(k)
501      format (1x,f5.0,6(1x,f10.3))
C
        go to 222
   333 continue
        close(22)
        close(23)
c
c ISEQ: sequntial number in the current file
c CRUISE: NODC cruise number
c STATNR: internal station number
c COUNTRY: NODC country code
c PLAT:NODC platform code
```

```
c XLON: longitude in decimal degrees
c XLAT: latitude in decimal degrees
c IDAY: day of observation
c IMON: month of observation
c IYEAR: year of observation
c IHOUR: hour of observation
c IMIN: minute of observation
c IBOTTD: bottom depth
c IMAXOD: maximum observed depth
c NOBS: number of observed depths
c DATTYP: observation type
c CDIR: direction of cast (only CTD-data)
c IWMOSQ: ten degree WMO square number
c data are ordered in the following case:
c DEPTH, TEMPERATURE, SALINITY, OXYGENE, PHOSPHATE, SILICATE, NITRATE
C
      stop '***END***'
       end
```

Inter - 11

```
      program AWIDMI
C
C     Guretsky, AWI, June 1991
C
C/CONVERTION OF DECIBARS INTO METERS (OBSERVED LEVELS AND MAX_OBS_DEPTH)
C
C/INTERPOLATION AT STANDARD LEVELS
C
      real p(9000),PMAX,z(9000)
      character*20 Cruise
      character file1*30, file2*30
      integer*4 Crnumber
      real*4    zg1(5000),tg1(5000),sg1(5000),zst(42),
     *       fob1(5000), zob1(5000) ,TST(42),SST(42)
C
C THESE ARE THE STANDARD LEVELS OF THE DATA_BASE:
      data zst /0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,
     * 250.,300.,350.,400.,500.,600.,700.,750.,800.,900.,
     * 1000.,1100.,1200.,1300.,1400.,1500.,1750.,2000.,2250.,2500.,
     * 2750.,3000.,3250.,3500.,3750.,4000.,4500.,5000.,5500.,6000.,
     * 6500.,7000./
C-------------------------------------------------------------------
      type*,'name of the input file'
      accept 100, file1
      open(unit=21, file=file1
     *,status='old')
C (INPUT FILES ARE:
C   ANT2.DAT ANT3.DAT ANT5.DAT ANT51.DAT ANT7.DAT ANT8.DAT)
      type*,'name of the output '
      accept 100, file2
      open(unit=22, file=file2
     *,status='new')
C OUTPUT FILES ARE: ANT2I.DAT ANT3I.DAT ANT5I.DAT ANT51I.DAT ANT7I.DAT
C ANT8I.DAT
      type*,'insert Cruise_Number (WILL BE ASSIGNED TO ALL STATIONS
     * OF THE SAME CRUISE'
      accept*, Crnumber
C CRUISE NUMBERS ARE 59001 59002 59003 59004 59005 59006 59007
C-------------------------------------------------------------------
  222 continue
C READ HEADER
      read(21,*,end=333) ISTAT
      type*,istat
      read(21,*) NGRADP,AMINP,NGRADL,AMINL
      read(21,*) NDA,MON,NYE,NHO,MIN
      read(21,*) MBDEPTH
      read(21,*) MLAST !!LAST OBSERVED LEVEL IN DBAR
      read(21,*)KMAX
C
C READ OBSERVED LEVELS
      do 5 k=1,KMAX
      read(21,*)IN,P(k),TG1(k),SG1(k)
      call condbar(p(k),PH,zG1(k))  !! CONVERSION
    5 continue
C
C GIVE FULL YEAR
      NYE=NYE+1900
C
C CONVERT MINUTES
      dp=ABS(AMINP)/60.
      if(NGRADP.LT.0)GRADP=float(NGRADP)-DP
      if(NGRADP.GE.0)GRADP=float(NGRADP)+DP
C
      dl=ABS(AMINL)/60.
      if(NGRADL.lt.0) GRADL=float(NGRADL)-DL
      if(NGRADL.GE.0) GRADL=float(NGRADL)+DL
```

```
C
C CONVERT LAST OBSERVED LEVEL
      flast=float(MLAST)
      call condbar(flast,PH,ZLAST)
      izlast=ifix(ZLAST)
C-------------------------------------------------------------------
C
C     I N T E R P O L A T I O N   A T   S T A N D A R D   L E V E L S
C SET DUMMY
      do 347 kk=1,42
      TST(kk)=-99.9
      sst(kk)=-99.9
  347 continue
C
C SET ALLOWED RANGE OF TEMPERATURE
      fmin=-2.3
      fmax=29.
      NZ = KMAX
CCC      type*,'nz=',nz
      mt=inter(nz, zg1, tg1, fmin, fmax, TST, zst, nob2, fob1, zob1)
C
C SET ALLOWED RANGE OF SALINITY
      fmin=10.
      fmax=36.5
      ms=inter(nz, zg1, sg1, fmin, fmax, SST, zst, nob2, fob1, zob1)
C
      mm=mt
      if(ms.gt.mt)mm=ms

C         O U T P U T
C=-==============================================================
C      set values for the OCEAN surface if possible
C
      if(zg1(1).gt.0..and.zg1(1).lt.8.) TST(1)=Tg1(1)
      if(zg1(1).gt.0..and.zg1(1).lt.8.) SST(1)=sg1(1)
C
C RANGE CHECK OF THE INTERPOLATED DATA
      j=0
      do 8 i=1,mm
      if((tst(i).gt.30..or.tst(i).lt.-2.5).and.(sst(i).lt.20..or.
     *sst(i).gt.37.)) go to 8
   22 format(2x,i3,2x,f6.1,f7.3,f7.3)
      j=j+1
      zg1(j)=zst(i)
      tg1(j)=tst(i)
      sg1(j)=sst(i)
CCC      type*,J,zg1(j),tg1(j),sg1(j)
    8 continue
      NSTAND=J
C=================================================================
      MSQ=-999 !! SET DUMMY MARSDEN SQUARE
      NSEQ=NSEQ+1
      write(22,*) NSEQ ! seq number in the file
      write(22,*) Crnumber ! Cruise_Number
      write(22,*) ISTAT         ! station number
      write(22,*) GRADP, GRADL  ! Latitude, Longitude (grad)
      write(22,*) NDA,MON,NYE,NHO,MIN ! day, Month, Year, Hour, Min
      write(22,*) MBDEPTH, IZLAST ! Bottt_Depth (m) Max_Obs Depth (m)
      write(22,*) KMAX , Nstand! Number of Obs_Levels and Stand_Levels
      write(22,*) MSQ ! Marsden square
C
      do 9 k=1,NSTAND
      write(22,*) K, ZG1(k), TG1(k), SG1(k)  ! NUM  LEVEL  TEM  SAL
    9 continue
C========================================
      go to 222
```

```
333 continue
100 format(a30)
    close(unit=22)
    close(unit=20)
    stop '********* E N D *********'
    END
```

Inter-10

```
      program interarg
C V.Guretsky, AWI, August 1991
C
      real*4 zz(80),tem(80), sal(80), oxy(80), ongitud, atitud,
     *       tst(42), sst(42), oxst(42), zst(42), fob1(80), zob1(80),
     *       PO(80),N3(80),SI(80),POST(42),N3ST(42),SIST(42)
C
      character file1*15, file2*15
      integer*4 NCRUISE
C
      data zst/0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,250.,
     *300.,350.,400.,500.,600.,700.,750.,800.,900.,1000.,1100.,
     *1200.,1300.,
     *1400.,1500.,1750.,2000.,2250.,2500.,2750.,3000.,3250.,3500.,
     *3750.,4000.,4500.,5000.,5500.,6000.,6500.,7000./
C
      type*,'input file'
      accept 100, file1
  100 format(a15)
      open(unit=22,file=file1,status='old')
C
      type*,'outputfile'
      accept 100, file2
      open(unit=23, file=file2, status='new')
      mseq=0
  222 continue
C*********************************************
C        I N P U T
      read(22,502,end=333) nseq,NCRUISE,nstat, ongitud,atitud
      read(22,503) nyear,nmonth,nday,
     *nhour,nmin,ndepth,modepth,KLEVEL,msq10
      read(22,504)country
      read(22,504)ship
      read(22,505)cruise
  502 format(2x,3i7,2f8.2)
  503 format(10i7)
  504 format(2x,a2)
  505 format(2x,a3)
      do kk=1,KLEVEL
      read(22,603) zz(kk), tem(kk), sal(kk),oxy(kk)
  603 format(2x,f5.0,2f7.3,f6.2)
      end do
      modepth=zz(KLEVEL)
C    ________________________________________________
c
      do 7 k=1,42
      sst(k)=0.
      oxst(k)=0.
      tst(k)=0.
      post(k)=0.
      n3st(k)=0.
      sist(k)=0.
    7 continue
C
      nz=KLEVEL
C
      fmin=-2.3
      fmax=29.
      mt=inter(nz,zz,tem,fmin,fmax,tst,zst,nob2,fob1,zob1)
      fmin=20.
      fmax=36.5
      ms=inter(nz,zz,sal,fmin,fmax,sst,zst,nob2,fob1,zob1)
      fmin=1.
      fmax=15.
      mox=inter(nz,zz,oxy,fmin,fmax,oxst,zst,nob2,fob1,zob1)
C
```

```
      mmax=mt
      if(ms.gt.mt)mmax=ms
      type*,'mseq=',nseq,'   mmax=',mmax,'   nz=',nz
  103 format(2x,f5.0,6f8.3)
  202 format(2x,3i7,2f8.2)
  212 format(2x,9i7)
c
      write(23,202) nseq,NCRUISE,nstat,ongitud,atitud
      write(23,212)nyear,nmonth,nday,
     *nhour,nmin,ndepth,modepth,KLEVEL,msq10
      write(23,102) mmax
  102 format(2x,i3)
      do 11 k=1, mmax
   11 write(23,103) zst(k), tst(k), sst(k), oxst(k)
C
      type*,nseq
      go to 222
  333 continue
      close (unit=23)
      close(unit=22)
      type*,'MSEQ=',nseq
      stop '***END***'
       end
```

Muinter
June 91

```
      program MUINTER
C     interpolation  to the standard depths.
C
C     V.Guretsky, AWI, JUNE 1991
C
      character file1*15, file2*15
C
      integer*4 NCRU
C
      real*4    zg1(5000),tg1(5000),sg1(5000),zst(42),
     *       fob1(5000), zob1(5000) ,TST(42),SST(42)
C
      data zst /0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,
     * 250.,300.,350.,400.,500.,600.,700.,750.,800.,900.,
     * 1000.,1100.,1200.,1300.,1400.,1500.,1750.,2000.,2250.,2500.,
     * 2750.,3000.,3250.,3500.,3750.,4000.,4500.,5000.,5500.,6000.,
     * 6500.,7000./
C     -------------------------------------
  100 format(a15)
C
       type*, 'Name of input file'
       accept 100,file1
       open(unit=20, file=file1,status='old')
C
       type*, 'Name of output file'
       accept 100, file2
       open(unit=22, file=file2,status='new')
C-----------------------------------------------
      mseq=0
  222 continue
C    __________________________________________________
      read(20,202,end=333) nseq,NCRU,numst, ongitud,atitud
      read(20,203) nyear,nmonth,nday,
     *nhour,nmin,ndepth,modepth,nlev,msq
      type*,'--R'
      type203, nyear,nmonth,nday,nhour,nmin
      type203,ndepth,modepth,nlev
C
C
      mseq=mseq+1
      type*,'--W'
      write(22,202) mseq,NCRU,numst, ongitud,atitud
      type203,nyear,nmonth,nday,nhour,nmin
      type203,ndepth,modepth,nlev
      write(22,203) nyear,nmonth,nday,
     *nhour,nmin,ndepth,modepth,nlev,msq
C
C
      read(20,104) (zg1(k), tg1(k), sg1(k),k=1,nlev)
  104 format(5(1x,f7.2,2f7.3))
  202 format(2x,3i7,2f8.2)
  203 format(10i7)
C------------------------------------------------------

C
C     I N T E R P O L A T I O N
      do 347 kk=1,42
      TST(kk)=-99.9
      sst(kk)=-99.9
  347 continue
C
      fmin=-2.3
      fmax=29.
      NZ = NLEV
      type*,'nz=',nz
      mt=inter(nz, zg1, tg1, fmin, fmax, TST, zst, nob2, fob1, zob1)
```

Inter 9

```
      fmin=10.
      fmax=36.5
      ms=inter(nz, zg1, sg1, fmin, fmax, SST, zst, nob2, fob1, zob1)
C
      mm=mt
      if(ms.gt.mt)mm=ms
C
C          O U T P U T
C=-====================================================
C      set values for the upper surface if possible
C
      if(zg1(1).gt.0..and.zg1(1).lt.3.) TST(1)=Tg1(1)
      if(zg1(1).gt.0..and.zg1(1).lt.3.) SST(1)=sg1(1)
C
C
C
      j=0
      do8 i=1,mm
      if((tst(i).gt.30..or.tst(i).lt.-2.5).and.(sst(i).lt.20..or.
     *sst(i).gt.37.)) go to 8
   22 format(2x,i3,2x,f6.1,f7.3,f7.3)
      j=j+1
      zg1(j)=zst(i)
      tg1(j)=tst(i)
      sg1(j)=sst(i)
    8 continue
C
      write(22,22) J
C
      do9 i=1,J
      ii=i
      write(22,22) ii, zg1(i),Tg1(i),Sg1(i)
    9 continue
C=========================================================
      go to 222
  333 continue
      close(unit=22)
      close(unit=20)
      stop '********* E N D *********'
      END
```

Interpolation of Gordon Data
Intergor
June 90

```
      program intergor
C     This program select data from the Standard_data table
C     for the specified Gordon Station  and makes
C     interpolation to the standard depths.
C
C     V.Guretsky, AWI, June 1990
      EXTERNAL err_handler
      External msg_handler
      include '(fsybdb) '
C
      Integer*4  dbproc, login,return_code,error,idg
C
       character file1*15, file2*15
C
      real*8 T8,O8,S8,Z8
      real*4 temg(42),salg(42),oxyg(42),
     *       zg1(80),tg1(80),sg1(80),og1(80),zst(42),
     *       fob1(80), zob1(80)
C
      login = fdblogin()
      call fdbsetluser(login,'SOCEAN')
      call fdbsetlpwd(login, 'Victor')
      dbproc = fdbopen(login, NULL)
      call fdbuse(dbproc,'SouthernOceanDB')
      data zst /0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,
     * 250.,300.,350.,400.,500.,600.,700.,750.,800.,900.,
     * 1000.,1100.,1200.,1300.,1400.,1500.,1750.,2000.,2250.,2500.,
     * 2750.,3000.,3250.,3500.,3750.,4000.,4500.,5000.,5500.,6000.,
     * 6500.,7000./
C
C     -----------------------------------
  100 format(a15)
C
       type*, 'Name of output file'
       read(6,100)file2
       open(unit=22, file=file2,status='new')
C
C
C  Selection of standard data for the gordon data
C
      do 222 i=1, 6313
      IDG=i+100000
      call fdbsetnull(dbproc,flt8bind,0,99.)
      call fdbfcmd(dbproc,'Execute Stadata %d', IDG)
      call fdbsqlexec(dbproc)
      call fdbresults(dbproc)
      call fdbbind(dbproc,1,flt8bind,0,Z8)
      call fdbbind(dbproc,2,flt8bind,0,T8)
      call fdbbind(dbproc,3,flt8bind,0,S8)
      call fdbbind(dbproc,4,flt8bind,0,O8)
      m=0
      do while(fdbnextrow(dbproc).ne.NO_MORE_ROWS)
      m=m+1
      zg1(m)=sngl(Z8)
      tg1(m)=sngl(T8)
      sg1(m)=sngl(S8)
      Og1(m)=sngl(O8)
      end do
C
      do 11 k=1,42
      temg(k)=0.
      salg(k)=0.
      oxyg(k)=0.
   11 continue
C
C     INTERPOLATION OF GORDON DATA
```

Inter-8

```
C
C     I N T E R P O L A T I O N
      fmin=-2.3
      fmax=29.
      mt=inter(m, zg1, tg1, fmin, fmax, temg, zst, nob2, fob1, zob1)
      fmin=10.
      fmax=36.5
      ms=inter(m, zg1, sg1, fmin, fmax, salg, zst, nob2, fob1, zob1)
      fmin=1.
      fmax=14.
      mox=inter(m, zg1, og1, fmin, fmax, oxyg, zst, nob2, fob1, zob1)
C
      mmax=max0(mt,ms,mox)
C
      type*,i,IDG
      write(22,99) i, IDG, mmax
      do 20 k=1,mmax
      write(22,300)zst(k), temg(k), salg(k), oxyg(k)
   20 continue
  222 continue
   99 format(2x,i4,2x,i7,2x,i2)
  300 format(2x,f5.0,3(2x,f8.4))
      close(unit=22)
      call fdbexit()
      stop '********* E N D *********'
      END
C
C
C     Error und Message Handler fuer
C     embedded SQL-Programme. In diesen mit
C     INCLUDE '(ERRMSG)' includen.
C
C     Error Handler
C     -------------
C     ERR_HANDLER - This funtion may be coded within the same program
C     or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION err_handler (dbproc, severity, errno, oserrno)
C
       include '(fsybdb)'
C
C      EXTERNAL         err_handler
C      EXTERNAL         msg_handler
C
      INTEGER*4        dbproc
      INTEGER*4        severity
      INTEGER*4        errno
      INTEGER*4        oserrno
      INTEGER*4        length
      INTEGER*4        return_code
C
      CHARACTER*(80)   message
C
         length = fdberrstr(errno,message)
         type *, 'DB-LIBRARY error: ', message
C
C     Check for operating system errors
C
         length = 0
         message = ' '
         length = fdboserrstr(oserrno, message)
C
         if (oserrno .ne. DBNOERR) then
            type *, 'Operating-system error: ', message
         end if
C
```

```
         return_code = fdbdead(dbproc)
C
         if ((dbproc .eq. NULL) .OR. (return_code ) .OR.
     2      (severity .eq. EXSERVER)) then
            err_handler = INT_EXIT
C
         else
            err_handler = INT_CANCEL
         end if
C
         END
C
C       Message Handler
C       ---------------
C MSG_HANDLER - This funtion may be coded within the same program
C               or as a separate file that is compiled/linked.
C
      INTEGER*4 FUNCTION msg_handler (dbproc, msgno,
     2             msgstate,severity, msgtext)
C
      include '(fsybdb)'
C
      INTEGER*4      dbproc
      INTEGER*4      msgno
      INTEGER*4      msgstate
      INTEGER*4      severity
C
      CHARACTER*80   msgtext
        IF (MSGNO.NE.5701) THEN
C
         type *, 'DataServer message ', msgno,
     2      '   state ', msgstate, '    severity ',
     3      severity,' ', msgtext
C
        END IF
         msg_handler = DBNOSAVE
C
      END
```

Intergon
December 90

```
      program intergon
C     interpolation of GONELLA' s data to the standard depths.
C
C     V.Guretsky, AWI, 13 DECEMBER 1990
C
      character*20 Cruise
       character file1*15, file2*15
C
      real*4    zg1(5000),tg1(5000),sg1(5000),og1(5000),zst(42),
     *       fob1(1000), zob1(5000) ,TST(42),SST(42),OST(42)
C
      data zst /0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,
     * 250.,300.,350.,400.,500.,600.,700.,750.,800.,900.,
     * 1000.,1100.,1200.,1300.,1400.,1500.,1750.,2000.,2250.,2500.,
     * 2750.,3000.,3250.,3500.,3750.,4000.,4500.,5000.,5500.,6000.,
     * 6500.,7000./
C     ------------------------------------
  100 format(a15)
C
       type*, 'Name of input file'
       read(6,100)file1
       open(unit=20, file=file1,status='old')
C
       type*, 'Name of output file'
       read(6,100)file2
       open(unit=22, file=file2,status='new')
C------------------------------------------------
  222 continue
      read(20,25,end=333) Cruise, nstat, ALA, PHI, ndepth,amaod,
     * nyear, month,
     *nday, TIME,NZ
C
      do i=1,NZ
      read(20,22)j,zg1(j),tg1(j),sg1(j),Og1(j)
      end do
C
   25 format(2x,a20,1x,i4,1x,f9.4,1x,f9.4,1x,i4,1x,f6.1,1x,i4,1x,
     *i2,1x,
     *i2,1x,
     *f3.0,1x,i3)
   22 format(2x,i3,2x,f6.1,f7.3,f7.3,f6.2)
C-----------------------------------------------------------------
C     INTERPOLATION
C
C     I N T E R P O L A T I O N
      do 347 kk=1,42
      TST(k)=-99.9
      sst(k)=-99.9
  347 OST(k)=-99.9
C
      fmin=-2.3
      fmax=29.
      mt=inter(nz, zg1, tg1, fmin, fmax, TST, zst, nob2, fob1, zob1)
      fmin=10.
      fmax=36.5
      ms=inter(nz, zg1, sg1, fmin, fmax, SST, zst, nob2, fob1, zob1)
      fmin=1.
      fmax=14.
      mox=inter(nz, zg1, og1, fmin, fmax, OST, zst, nob2, fob1, zob1)
C
      mmax=max0(mt,ms,mox)
C
C          O U T P U T
C=-================================================
C      set values for the upper surface if possible
C
```

Inter-7

```
       if(zg1(1).gt.0..and.zg1(1).lt.3.) TST(1)=Tg1(1)
       if(zg1(1).gt.0..and.zg1(1).lt.3.) SST(1)=sg1(1)
       if(zg1(1).gt.0..and.zg1(1).lt.3.) OST(1)=Og1(1)
C
       M=M+1
       type*,M
C
       write(22,25) Cruise, nstat, ALA, PHI, ndepth,amaod,
      * nyear, month,
      *nday, TIME,NZ
C
       write(22,22)mmax
       do i=1,mmax
       write(22,22) i, zst(i),TST(i),SST(i),OST(i)
       end do
C=============================================================
       go to 222
   333 continue
       close(unit=22)
       close(unit=20)
       stop '********* E N D *********'
       END
```

Interheinz
November 90

```
      program interheinz
C     interpolation of heinz s data to the standard depths.
C     changes max_obs_pressure for max_obse_depth
C
C     V.Guretsky, AWI, November 1990
C
C
       character file1*15, file2*15
C
      real*4    zg1(900),tg1(900),sg1(900),og1(900),zst(42),
     *       fob1(900), zob1(900) ,TST(42),SST(42),OST(42)
C
      integer*4 CRUNU
C
      data zst /0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,
     * 250.,300.,350.,400.,500.,600.,700.,750.,800.,900.,
     * 1000.,1100.,1200.,1300.,1400.,1500.,1750.,2000.,2250.,2500.,
     * 2750.,3000.,3250.,3500.,3750.,4000.,4500.,5000.,5500.,6000.,
     * 6500.,7000./
C
C     -------------------------------------
  100 format(a15)
C
      msq=0
       type* ,'insert new Cruise_Number I6'
       accept901,CRUNU
  901 format(I6)
C
       type*, 'Name of input file'
       read(6,100)file1
       open(unit=21, file=file1,status='old')
C
       type*, 'Name of output file'
       read(6,100)file2
       open(unit=22, file=file2,status='new')
C
C
C
  222 continue
      read(21,101,end=333) nseq, ppcc, ns, ongitud, atitud,NYEAR,NMO
      read(21,111) NDA,NHO,NDE,PMAX,NZ,SYMBOL,CRUISE
      idum=0
      do 1 k=1,nz
      read(21,121) zg1(k), tg1(k), sg1(k), og1(k)
      if(og1(k).gt.20..and.og1(k).lt.80.) idum=1
    1 continue
      if(idum.eq.0) go to 345
      do 346 k=1,nz
  346 og1(k)=-99.9
C
C     nseq - sequential number of station in the file
C     ppcc - NODC platform code, NODC country code of the platform
C     ns - station_number
C     ongitud - Longitude
C     atitude - Latitude
C     nyear - Year
C     nmo - month
C     nda - day
C     nho - hour
C     nde - Bottom_Depth
C     mod - Max_Obse_pressure bzw. _depth
C     nz - number_obse
C
C
  101 format(2X,i10,a5,i10,E12.5e2,E12.5e2,2I10)
  111 FORMAT(3i10,E12.5e2,i10,a1,a13)
```

Inter-6

```
  121 FORMAT(4E12.5E2)
  201 format(2x,i6,2x,a5,i7,2x,2f8.2,5i7,F8.2,i7,2X,a13)
  102 format(2x,f7.2,1x,3f8.3)
C
  345 continue
C     INTERPOLATION
C
C     I N T E R P O L A T I O N
      do 347 k=1,42
      TST(k)=-99.9
      sst(k)=-99.9
  347 OST(k)=-99.9
C
      fmin=-2.3
      fmax=29.
      mt=inter(nz, zg1, tg1, fmin, fmax, TST, zst, nob2, fob1, zob1)
      fmin=10.
      fmax=36.5
      ms=inter(nz, zg1, sg1, fmin, fmax, SST, zst, nob2, fob1, zob1)
      fmin=1.
      fmax=14.
      mox=inter(nz, zg1, og1, fmin, fmax, OST, zst, nob2, fob1, zob1)
C
      mmax=max0(mt,ms,mox)
C
C          O U T P U T
      M=M+1
      Phi=ABS(atitud)
      if(Pmax.lt.0..or.Pmax.gt.7500.)Pmax=0
      call convertdbar(PMAX,Phi,ZZZ)
      mod=ZZZ
      if(Pmax.eq.0.)mod=zg1(nz)
      type*,M, mod
      write(22,401) nseq, CRUNU, ns, ongitud, atitud, nyear, nmo, nda,
     * nho,
     * nde, mod, nz, msq
      write(22,401)mmax
      do 11 k=1,mmax
   11 write(22,102) zst(k), tst(k), sst(k), OST(K)
  401 format(2x,3I7,2x,2f9.4,2x,8i5)
      go to 222
C
  333 continue
   99 format(2x,i4,2x,i7,2x,i2)
  300 format(2x,f5.0,3(2x,f8.4))
      close(unit=22)
      stop '********* E N D *********'
      END
```

Interowl
August 90

```
      program internowl
C V.Guretsky, AWI, August 1990
C
      real*4 z(80),tem(80), sal(80), oxy(80), ongitud, atitud,
     *       tst(42), sst(42), oxst(42), zst(42), fobl(80), zobl(80)
C
      character file1*15, file2*15
C
      data zst/0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,250.,
     *300.,350.,400.,500.,600.,700.,750.,800.,900.,1000.,1100.,
     *1200.,1300.,
     *1400.,1500.,1750.,2000.,2250.,2500.,2750.,3000.,3250.,3500.,
     *3750.,4000.,4500.,5000.,5500.,6000.,6500.,7000./
C
      type*,'input file'
      accept 100, file1
  100 format(a15)
      type*,'outputfile'
      accept 100, file2
      open (unit=21, file=file1, status='old')
      open(unit=22, file=file2, status='new')
  222 continue
      read(21,102,end=333) nseq,nc,ns,ongitud,atitud,nye,nmo,nda,nho,
     *nde,mod,nz,msq
C
      do 1 k=1,nz
    1 read(21,103) z(k), tem(k), sal(k), oxy(k)
C
  102 format(2x,3i7,2f8.2,9i6)
  103 format(2x,f5.0,3f8.3)
c
      do 7 k=1,42
      sst(k)=0.
      oxst(k)=0.
      tst(k)=0.
    7 continue
C
      fmin=-2.3
      fmax=29.
      mt=inter(nz,z,tem,fmin,fmax,tst,zst,nob2,fob1,zob1)
      fmin=20.
      fmax=36.5
      ms=inter(nz,z,sal,fmin,fmax,sst,zst,nob2,fob1,zob1)
      fmin=1.
      fmax=15.
      mox=inter(nz,z,oxy,fmin,fmax,oxst,zst,nob2,fob1,zob1)
C
      mmax=max0(mt,ms,mox)
c
      write(22,102) nseq, nc, ns, ongitud, atitud, nye, nmo,nda,nho,
     *nde, mod, nz, msq
C
C      WE ASSUME HERE that 1-meter level is the same for the surface
      if(z(1).le.1.)go to 33
       go to 34
   33 tst(1)=tem(1)
      sst(1)=sal(1)
      oxst(1)=oxy(1)
   34 continue
C
      write(22,102) mmax
      do 11 k=1, mmax
   11 write(22,103) zst(k), tst(k), sst(k), oxst(k)
C
      go to 222
  333 continue
```

Inter-5

```
close (unit=21)
close(unit=22)
type*,'NSEQ=',nseq
stop '***END***'
 end
```

Interpol
August 90

```
      program interpol
C V.Guretsky, AWI, August 1990
C
      real*4 z(80),tem(80), sal(80), oxy(80), ongitud, atitud,
     *       tst(42), sst(42), oxst(42), zst(42), fob1(80), zob1(80)
C
      character file1*15, file2*15
C
      data zst/0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,250.,
     *300.,350.,400.,500.,600.,700.,750.,800.,900.,1000.,1100.,
     *1200.,1300.,
     *1400.,1500.,1750.,2000.,2250.,2500.,2750.,3000.,3250.,3500.,
     *3750.,4000.,4500.,5000.,5500.,6000.,6500.,7000./
C
      type*,'input file'
      accept 100, file1
  100 format(a15)
      type*,'outputfile'
      accept 100, file2
      open (unit=21, file=file1, status='old')
      open(unit=22, file=file2, status='new')
  222 continue
      read(21,102,end=333) nseq,nc,ns,ongitud,atitud,nye,nmo,nda,nho,
     *nde,mod,nz,msq
C
      do 1 k=1,nz
    1 read(21,103) z(k), tem(k), sal(k), oxy(k)
C
  102 format(2x,3i7,2f8.2,9i7)
  103 format(2x,f5.0,3f8.3)
c
      do 7 k=1,42
      sst(k)=0.
      oxst(k)=0.
      tst(k)=0.
    7 continue
C
      fmin=-2.3
      fmax=29.
      mt=inter(nz,z,tem,fmin,fmax,tst,zst,nob2,fob1,zob1)
      fmin=20.
      fmax=36.5
      ms=inter(nz,z,sal,fmin,fmax,sst,zst,nob2,fob1,zob1)
      fmin=1.
      fmax=15.
      mox=inter(nz,z,oxy,fmin,fmax,oxst,zst,nob2,fob1,zob1)
C
      mmax=max0(mt,ms,mox)
c
      write(22,102) nseq, nc, ns, ongitud, atitud, nye, nmo,nda,nho,
     *nde, mod, nz, msq
      write(22,102) mmax
      do 11 k=1, mmax
   11 write(22,103) zst(k), tst(k), sst(k), oxst(k)
C
      go to 222
  333 continue
      close (unit=21)
      close(unit=22)
      type*,'NSEQ=',nseq
      stop '***END***'
       end
```

Inter-4

Interjap1
August-90

Tokyo
Interpolation of Standard-levels

```
      program interjap1
C V.Guretsky, AWI, August 1990
C
      real*4 zz(80),tem(80), sal(80), oxy(80), ongitud, atitud,
     *       tst(42), sst(42), oxst(42), zst(42), fobl(80), zobl(80)
C
      character file1*15, file2*15
      integer*2 crunu
C
      data zst/0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,250.,
     *300.,350.,400.,500.,600.,700.,750.,800.,900.,1000.,1100.,
     *1200.,1300.,
     *1400.,1500.,1750.,2000.,2250.,2500.,2750.,3000.,3250.,3500.,
     *3750.,4000.,4500.,5000.,5500.,6000.,6500.,7000./
C
      type*,'input file'
      accept 100, file1
  100 format(a15)
      open(unit=21,file=file1,status='old')
C
      type*,'outputfile'
      accept 100, file2
      open(unit=22, file=file2, status='new')
      mseq=0
  222 continue
C*********************************************
C         I N P U T
      read(21,202,end=333) nseq,CRUNU,numstat,A,P,nyear,month,nday,
     *nhour,minut,ndep,modepth,n,msq
      do 2 k=1,n
    2 read(21,103) zz(k), tem(k), sal(k), oxy(k)
  103 format(2x,f5.0,3f8.3)
  202 format(2x,3i7,2f8.2,9i7)
C    ______________________________________________
c
      do 7 k=1,42
      sst(k)=0.
      oxst(k)=0.
      tst(k)=0.
    7 continue
C
      nz=n
C
      fmin=-2.3
      fmax=29.
      mt=inter(nz,zz,tem,fmin,fmax,tst,zst,nob2,fobl,zobl)
      fmin=20.
      fmax=36.5
      ms=inter(nz,zz,sal,fmin,fmax,sst,zst,nob2,fobl,zobl)
      fmin=1.
      fmax=15.
      mox=inter(nz,zz,oxy,fmin,fmax,oxst,zst,nob2,fobl,zobl)
C
CCC      mmax=max0(mt,ms,mox)
      mmax=mt
      if(ms.gt.mt)mmax=ms
      type*,'mseq=',mseq,'   mmax=',mmax,'   nz=',nz
c
      Mseq=Mseq+1
      write(22,202) mseq,CRUNU,numstat,A,P,nyear,month,nday,
     *nhour,minut,ndep,modepth,n,msq
      write(22,102) mmax
  102 format(2x,i3)
      do 11 k=1, mmax
   11 write(22,103) zst(k), tst(k), sst(k), oxst(k)
C
```

Inter-3

```
      go to 222
  333 continue
      close (unit=21)
      close(unit=22)
      type*,'MSEQ=',mseq
      stop '***END***'
       end
```

Inkejap2
August 90

```
      program interjap2        (Jae)
C V.Guretsky, AWI, August 1990
C
      real*4 zz(80),tem(80), sal(80), oxy(80), ongitud, atitud,
     *      tst(42), sst(42), oxst(42), zst(42), fobl(80), zobl(80),
     *      PO(80),N3(80),SI(80),POST(42),N3ST(42),SIST(42)
C
      character file1*15, file2*15
      integer*2 crunu
C
      data zst/0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,250.,
     *300.,350.,400.,500.,600.,700.,750.,800.,900.,1000.,1100.,
     *1200.,1300.,
     *1400.,1500.,1750.,2000.,2250.,2500.,2750.,3000.,3250.,3500.,
     *3750.,4000.,4500.,5000.,5500.,6000.,6500.,7000./
C
      type*,'input file'
      accept 100, file1
  100 format(a15)
      open(unit=21,file=file1,status='old')
C
      type*,'outputfile'
      accept 100, file2
      open(unit=22, file=file2, status='new')
      mseq=0
  222 continue
C*********************************************
C        I N P U T
      read(21,202,end=333) nseq,CRUNU,numstat,A,P,nyear,month,nday,
     *nhour,minut,ndep,modepth,n,msq
      do 2 k=1,n
    2 read(21,103) zz(k), tem(k), sal(k), oxy(k),PO(k),N3(k),SI(k)
  103 format(2x,f5.0,6f8.3)
  202 format(2x,3i7,2f8.2,9i7)
C   ______________________________________________
c
      do 7 k=1,42
      sst(k)=0.
      oxst(k)=0.
      tst(k)=0.
      post(k)=0.
      n3st(k)=0.
      sist(k)=0.
    7 continue
C
      nz=n
C
      fmin=-2.3
      fmax=29.
      mt=inter(nz,zz,tem,fmin,fmax,tst,zst,nob2,fobl,zobl)
      fmin=20.
      fmax=36.5
      ms=inter(nz,zz,sal,fmin,fmax,sst,zst,nob2,fobl,zobl)
      fmin=1.
      fmax=15.
      mox=inter(nz,zz,oxy,fmin,fmax,oxst,zst,nob2,fobl,zobl)
C
      fmin=0.
      fmax=10.
      mpo=inter(nz,zz,po,fmin,fmax,post,zst,nob2,fobl,zobl)
C
      fmin=0.
      fmax=100.
      mn3=inter(nz,zz,N3,fmin,fmax,N3ST,zst,nob2,fobl,zobl)
      fmin=0.
      fmax=200.
```

Inter2

```
      msi=inter(nz,zz,SI,fmin,fmax,SIst,zst,nob2,fob1,zob1)
      mmax=mt
      if(ms.gt.mt)mmax=ms
      type*,'mseq=',mseq,'   mmax=',mmax,'   nz=',nz
c
      Mseq=Mseq+1
      write(22,202) mseq,CRUNU,numstat,A,P,nyear,month,nday,
     *nhour,minut,ndep,modepth,n,msq
      write(22,102) mmax
  102 format(2x,i3)
      do 11 k=1, mmax
   11 write(22,103) zst(k), tst(k), sst(k), oxst(k),post(k),n3st(k),
     *sist(k)
C
      go to 222
  333 continue
      close (unit=21)
      close(unit=22)
      type*,'MSEQ=',mseq
      stop '***END***'
       end
```

Interjare
November 90

```
      program interjare
C     interpolation of JARE data to the standard depths.
C
C     V.Guretsky, AWI, November 1990
C
C
       character file1*15, file2*15
C
      real*4    zg1(900),tg1(900),sg1(900),og1(900),zst(42),
     *       fob1(900), zob1(900) ,TST(42),SST(42),OST(42),
     * ANI1(900),APH1(900),ASI1(900),ANIST(42),APHST(42),ASIST(42)
C
      integer*4 CRUNU
C
      data zst /0.,10.,20.,30.,50.,75.,100.,125.,150.,200.,
     * 250.,300.,350.,400.,500.,600.,700.,750.,800.,900.,
     * 1000.,1100.,1200.,1300.,1400.,1500.,1750.,2000.,2250.,2500.,
     * 2750.,3000.,3250.,3500.,3750.,4000.,4500.,5000.,5500.,6000.,
     * 6500.,7000./
C
C     ------------------------------------
  100 format(a15)
C
      msq=0
       type* ,'insert new Cruise_Number I6'
       accept901,CRUNU
  901 format(I6)
C
       type*, 'Name of input file'
       read(6,100)file1
       open(unit=21, file=file1,status='old')
C
       type*, 'Name of output file'
       read(6,100)file2
       open(unit=22, file=file2,status='new')
C
C
C
  222 continue
      read(21,101,end=333) nseq, ppcc, ns, ongitud, atitud,NYEAR,NMO
      read(21,111) NDA,NHO,NDE,PMAX,NZ,SYMBOL,CRUISE
CCC      idum=0
      do 1 k=1,nz
      read(21,121) zg1(k), tg1(k), sg1(k), og1(k),APH1(k),ANI1(k),
     *ASI1(k)
CCC      if(og1(k).gt.20..and.og1(k).lt.80.) idum=1
    1 continue
CCC      if(idum.eq.0) go to 345
CCC      do 346 k=1,nz
CCC  346 og1(k)=-99.9
C
C     nseq - sequential number of station in the file
C     ppcc - NODC platform code, NODC country code of the platform
C     ns - station_number
C     ongitud - Longitude
C     atitude - Latitude
C     nyear - Year
C     nmo - month
C     nda - day
C     nho - hour
C     nde - Bottom_Depth
C     mod - Max_Obse_pressure bzw. _depth
C     nz - number_obse
C
C
  101 format(2X,i10,a5,i10,E12.5e2,E12.5e2,2I10)
```

Inte-1

```
  111 FORMAT(3i10,E12.5e2,i10,a1,a13)
  121 FORMAT(4E12.5E2)
  201 format(2x,i6,2x,a5,i7,2x,2f8.2,5i7,F8.2,i7,2X,a13)
  102 format(2x,f7.2,1x,6f8.3)
C
  345 continue
C     INTERPOLATION
C
C     I N T E R P O L A T I O N
      do 347 k=1,42
      TST(k)=-99.9
      APHST(k)=-99.9
      ANIST(k)=-99.9
      ASIST(k)=-99.9
      sst(k)=-99.9
  347 OST(k)=-99.9
C
      fmin=-2.3
      fmax=29.
      mt=inter(nz, zg1, tg1, fmin, fmax, TST, zst, nob2, fob1, zob1)
C
      fmin=10.
      fmax=36.5
      ms=inter(nz, zg1, sg1, fmin, fmax, SST, zst, nob2, fob1, zob1)
C
      fmin=1.
      fmax=14.
      mox=inter(nz, zg1, og1, fmin, fmax, OST, zst, nob2, fob1, zob1)
C
      fmin=
      fmax=
      mph=inter(nz, zg1, APH1, fmin, fmax, APHST, zst, nob2, fob1, zob1)
C
      fmin=
      fmax=
      mni=inter(nz, zg1, ANI1, fmin, fmax, ANIST, zst, nob2, fob1, zob1)
      fmin=
      fmax=
      msi=inter(nz, zg1, ASI1, fmin, fmax, ASIST, zst, nob2, fob1, zob1)
C
      mmax=max0(mt,ms,mox,mph,mni,msi)
C
C
C==============================================
      write(22,401) nseq, CRUNU, ns, ongitud, atitud, nyear, nmo, nda,
     * nho,
     * nde, mod, nz, msq
      write(22,401)mmax
      do 11 k=1,mmax
   11 write(22,102) zst(k), tst(k), sst(k), OST(K),APHST(k),ANIST(k),
     *ASIST(k)
  401 format(2x,3I7,2x,2f9.4,2x,8i5)
C==============================================
      go to 222
C
  333 continue
   99 format(2x,i4,2x,i7,2x,i2)
  300 format(2x,f5.0,3(2x,f8.4))
      close(unit=22)
      stop '********* E N D *********'
      END
```

Inter

```
      function inter(nob, zob, fob, fmin, fmax, fst, zst,
     * nob2, fob1, zob1)
C
C       V.Guretsky, AWI, June, 1990
C
C     nob - initial number of observed levels (INPUT)
C     zst(nob) - array of standard levels     (INPUT)
C     zob(nob) - array of initial observed levels (INPUT)
C     fob(nob) - array of initial observed values (INPUT)
C     fmin, fmax -  min-max limits for the observed values (INPUT)
C
C     fst(80) - array of interpolated values                 (OUTPUT)
C     nob2 - final number of observed levels with good data (OUTPUT)
C     fob1(nob1) - array of obs. values within min-max limits (OUTPUT)
C     zob1(nob1) - observed levels with "good"  data          (OUTPUT)
C     mst=inter - number of stand. levels for which interpolation (OUTPUT)
C           has been done
C
C     this version  uses 42 standard depth levels from 0 to 7000 meters
      nst=42
C
C
C
      real*4 zst(80), fst(80), zob(5000), fob(5000), fob1(5000),
     * zob1(5000)
C
      enter(x,x1,x2,y1,y2) = y1+(x-x1)*(y2-y1)/(x2-x1)
C
      k=0

C       selection of levels with good data
      do 4 L=1,nob
      if(fob(L).gt.fmax.or.fob(L).lt.fmin) go to 4
      if(L.eq.nob) go to 44
      if(zob(L+1).le.zob(L)) go to 4
   44 k = k + 1
      fob1(k)=fob(L)
      zob1(k)=zob(L)
    4 continue
C
      nob2=k   ! this is number of levels with good data
      nob1=nob2-1
      if(nob2.eq.0) go to 99
      do 1 k=1, nst
         if(zst(k).gt.zob1(nob2)) go to 222
            do 2 L=1, nob2
            if(L.eq.nob2) go to 75
            if(zst(k).eq.zob1(L)) go to 65
            if(zst(k).gt.zob1(L).and.zst(k).lt.zob1(L+1)) go to 3
            go to 2
    3 continue
C
C     LINeAR INTERPOLATION
      fst(k) = enter(zst(k), zob1(L), zob1(L+1), fob1(L), fob1(L+1))
      mst=k
      go to 1
   65 fst(k)=fob1(L)
      mst=k
      go to 1
   75 if(zst(k)-zob1(nob2)) 55, 65,55
   55 fst(k)= -99.9
      go to 1
    2 continue
      go to 1
  222 fst(k)=-99.9
    1 continue
```

```
      go to 79
   99 continue
      do k=1,nst
      fst(k)=-99.
      end do
      mst=0
   79 inter=mst
      return
      end
```

MOPENDB.C
15.7.91

```
#module mopendb

/*
**++
**  FACILITY:
**
**      open database
**      close database
**
**  ABSTRACT:
**
**      This module contains a function which opens a database
**      and returns the dbproc to a SYBASE database.
**      The function is called with three parameters, none must be
**      specified. If any parameter is not specified the function
**      asks for the proper value.
**
**      The other function will close the access path to the databae.
**
**      Both functions will return NULL if in any case the open will
**      fail
**
**  AUTHORS:
**
**      Lutz-Peter Kurdelski
**
**
**  CREATION DATE:      1990-11-14
**
**  MODIFICATION HISTORY:
**
**--
*/


/*
**
**  INCLUDE FILES
**
*/

#include "getlog.h"
#include <sybfront.h>
#include <sybdb.h>
```

```
/*
**++
**  FUNCTIONAL DESCRIPTION:
**
**      DBPROCESS *opendb (char *, char *, char *)
**
**      This function accepts the databaseName, the username and
**      the password to open the database.
**
**      This function needs the module MGETLOG.OBJ which define
**      the needed basis i/o functions.
**
**  FORMAL PARAMETERS:
**
**      databaseName      * char [31]
**      username          * char [31]
**      password          * char [20]
**
**  IMPLICIT INPUTS:
**
**      none
**
**  IMPLICIT OUTPUTS:
**
**      none
**
**  FUNCTION VALUE:
**
**      NULL    if any operation fails
**      dbproc  if the open is successfull
**
**  SIDE EFFECTS:
**
**      none
**
**--
*/
DBPROCESS *opendb ( char * databaseName,
                    char * username,
                    char * password )
{
    struct llogrec *loginstruct;

    DBPROCESS *dbproc;
    LOGINREC *login;

    if ((loginstruct = getLLogin (databaseName, username, password)) == NULL)
    {
        return (NULL);
    }

/*
** open the database and use it
*/
    if ((login = dblogin()) == NULL)
    {
        return(NULL);
    }

    if ((DBSETLUSER(login,loginstruct->username) == FAIL) ||
        (DBSETLPWD(login,loginstruct->password) == FAIL))
    {
        return(NULL);
    }
    if ((dbproc = dbopen(login,NULL)) == NULL)
    {
        return(NULL);
    }
    if (dbuse(dbproc,loginstruct->databaseName) == FAIL)
    {

        return(NULL);
    }

    return(dbproc);
}
```

```
/*
**++
**  FUNCTIONAL DESCRIPTION:
**
**      exitdb ()
**
**  FORMAL PARAMETERS:
**
**      none
**
**  IMPLICIT INPUTS:
**
**      [@description or none@]
**
**  IMPLICIT OUTPUTS:
**
**      none
**
**  COMPLETION CODES:
**
**      none
**
**  SIDE EFFECTS:
**
**      none
**
**--
*/
void closedb ()
{
    dbexit();
}
```

STRING FUNCTIONS.

8.7.91

```
#module STRFUNC "String functions"

/*
**++
**  FACILITY:
**
**      Definitions of some special string functions
**      tobe used first in some loading programs for
**      SouithernOceanDB
**      [@tbs@]...
**
**  ABSTRACT:
**
**      [@tbs@]...
**
**  AUTHORS:
**
**      LutzPeter Kurdelski
**      Alfred-Wegener-Institute
**      for Polar and Marine Research
**      Am Handelshafen 12
**      D-2850 Bremerhaven
**      [@tbs@]...
**
**
**  CREATION DATE:      1991-06-10
**
**  MODIFICATION HISTORY:
**
**--
*/
/*[@include files@]*/
/*[@macro definitions@]*/

/*[@preprocessor directive@]...*/

/*[@data type or declaration@]...*/

char * strtlt (char *);

/*[@function definition@]...*/
```

```
/*
**++
**  FUNCTIONAL DESCRIPTION:
**
**      Kopiert einen String von Position start bis stop aus einem
**      anderen String heraus. Der neue String ist "\0", wenn die
**      Positionen nicht mit der Laenge des Eingabestrings vertraeg-
**      lich sind. Der kopierte string ist IMMER NULL-terminiert.
**
**      die Kopie ist immer von start bis stop EINSCHLIESSLICH,
**      d.h. strsub(s, 1, 1) kopiert genau EIN Zeichen.
**      [@tbs@]...
**
**  FORMAL PARAMETERS:
**
**      source  Sourcestring
**      start   Startposition
**      stop    Endposition (einschliesslich)
**      [@tbs@]...
**
**  IMPLICIT INPUTS:
**
**      none
**
**  IMPLICIT OUTPUTS:
**
**      Bereitstellung des Platzes fuer den String
**
**  FUNCTION VALUE:
**
**      Zeiger auf den kopierten String
**
**  SIDE EFFECTS:
**
**      none
**
**--
*/
char *strsub ( char * source, int start, int stop)
{
    static char * dest;
    int length;
/*    [@block declaration@]...*/

    length = strlen (source);

    if ((start <= length) && (stop <= length) && (start >= 0) && (stop >= 0))
    {
        if (stop < start)
        {
            length = stop;
            stop = start;
            start = length;
        }
        dest = (char *) malloc (stop - start + 2);
        for (length = 0;  start <= stop;)
        {
            dest [length++] = source [start++];
        }
        dest [length] = '\0';
    }
    else
    {
        dest = (char *) malloc (1);
        dest [0] = '\0';
        }
        return (dest);
}
```

```
/*
**++
**  FUNCTIONAL DESCRIPTION:
**
**      strtrr
**
**      truncated leading and trailing spaces and replace characters
**      from delimeter
**
**      Loescht in einem String Zwischenraumzeichen am Anfang und
**      Ende des Strings.
**      Ersetzt in einem String die in delim uebergebenen Sonder-
**      zeichen durch '_'
**      [@tbs@]...
**
**  FORMAL PARAMETERS:
**
**      source  Sourcestring
**      delim   Begrenzerzeichen
**      [@tbs@]...
**
**  IMPLICIT INPUTS:
**
**      none
**
**  IMPLICIT OUTPUTS:
**
**      Bereitstellung des Platzes fuer den String
**
**  FUNCTION VALUE:
**
**      Zeiger auf den gegebenenfalls neu generierten String
**
**  SIDE EFFECTS:
**
**      none
**
**--
*/
char * strtrr ( char * source, char * delim)
{
    char * dest;

    int i,
        j;

    dest = strtlt (source);

    for (i = 0; i < strlen(delim); i++)
        for (j = 0; j < strlen(dest); j++)
            if (dest[j] == delim[i])
                dest[j] = '_';
    return (dest);
}
```

```
/*
**++
**  FUNCTIONAL DESCRIPTION:
**
**      strtlt
**
**      truncate leading and trailing spaces
**
**      Entfernt aus einem String die Zeichen ' ', die
**      an Anfang und am Ende eines Strings vorhanden sind.
**      [@tbs@]...
**
**  FORMAL PARAMETERS:
**
**      char * source   der zu bearbeitende String
**      [@tbs@]...
**
**  IMPLICIT INPUTS:
**
**      none
**
**  IMPLICIT OUTPUTS:
**
**      none
**
**  FUNCTION VALUE:
**
**      Zeiger auf den modifizierten String (Kopie von source)
**      [@tbs@]...
**
**  SIDE EFFECTS:
**
**      none
**
**--
*/
char * strtlt (char * source)
{
    char * alpha,
         * omega,
         * dest;

    alpha = strchr(source,' ');
    omega = strrchr(source,' ');
    if (omega > alpha)
    {
        dest = (char *) malloc ((int) (omega - alpha) + 2);
        strcpy(dest,alpha,(int) (omega - alpha) + 1);
        dest [(int) (omega - alpha) + 1] = '\0';
    }
    else
    {
        dest = (char *) malloc (strlen(source) + 1);
        strcpy (dest, source);
    }
    return (dest);
}
```

DATABASE FUNCTION

10.6.91

```
#module DBFUNC "Database functions"

/*
**++
**  FACILITY:
**
**      Definitions of some special database functions
**      to be used first in some loading programs for
**      SouthernOceanDB
**      [@tbs@]...
**
**  ABSTRACT:
**
**      [@tbs@]...
**
**  AUTHORS:
**
**      LutzPeter Kurdelski
**      Alfred-Wegener-Institute
**      for Polar and Marine Research
**      Am Handelshafen 12
**      D-2850 Bremerhaven
**      [@tbs@]...
**
**
**  CREATION DATE:      1991-06-10
**
**  MODIFICATION HISTORY:
**
**--
*/
#include <sybfront.h>
#include <sybdb.h>
/*[@include files@]*/
/*[@macro definitions@]*/

/*[@preprocessor directive@]...*/

/*[@data type or declaration@]...*/

/*[@function definition@]...*/
```

```
/*
**++
**  FUNCTIONAL DESCRIPTION:
**
**      Bindet Daten aus einer Datenbank an eine Variable.
**
**  FORMAL PARAMETERS:
**
**      dbproc      DBPROC      Datenbankreferenz
**      question    char *      Anfrage an die Datanbank
**      idStat      void *      Zeiger auf eine Integer (Nummer)
**      binding     int         Datentype fuer die Antwort
**      len         DBINT       reservierter Platz fuer idStat

**  IMPLICIT INPUTS:
**
**      [@description or none@]
**
**  IMPLICIT OUTPUTS:
**
**      [@description or none@]
**
**  COMPLETION CODES:
**
**      none
**
**  SIDE EFFECTS:
**
**      none
**
**--
*/
 connectToDB (DBPROCESS * dbproc,
                char *question,
                void *idStat,
                int binding,
                DBINT len)
{
/*    [@block declaration@]...*/

    dbcmd(dbproc,question);
    dbsqlexec(dbproc);
    dbresults(dbproc);
    dbbind(dbproc,1,binding,len,idStat);
    dbnextrow(dbproc);

/*    [@statement@]...*/
}
```

mhandle
15.11.90

```
#module mhandler

#include <sybfront.h>
#include <sybdb.h>

int err_handler(dbproc, severity, dberr, oserr, dberrstr, oserrstr)
DBPROCESS        *dbproc;
int              severity;
int              dberr;
int              oserr;
char             *dberrstr;
char             *oserrstr;
{
        if ((dbproc == NULL) || (DBDEAD(dbproc)))
                return(INT_EXIT);
        else
        {
                printf("DB-Library error:\n\t%s\n", dberrstr);

                if (oserr != DBNOERR)
                        printf("Operating-system error:\n\t%s\n", oserrstr);

                return(INT_CANCEL);
        }
}

int msg_handler(dbproc, msgno, msgstate, severity, msgtext,
                srvname, procname, line)

DBPROCESS        *dbproc;
DBINT            msgno;
int              msgstate;
int              severity;
char             *msgtext;
char             *srvname;
char             *procname;
DBUSMALLINT      line;

{
      if (severity > 0)
      {
         printf ("Msg %ld, Level %d, State %d\n",
                 msgno, severity, msgstate);

         if (strlen(srvname) > 0)
                 printf ("Server '%s', ", srvname);
         if (strlen(procname) > 0)
                 printf ("Procedure '%s', ", procname);
         if (line > 0)
                 printf ("Line %d", line);

         printf("\n\t%s\n", msgtext);
       }
         return(0);
}
```

6C.C
14.11.90

```
#include curses
#define bool int
#define DBPROCESS int

typedef struct logrec {
   char benutzername[31],
        password[31];
    };

struct logrec *getLogin (char *, char *);
DBPROCESS *openDB(char *,char *);

main ()
{
    struct logrec *loginstruct;

    loginstruct = getLogin("test","");

    openDB(loginstruct->benutzername,loginstruct->password);
}

DBPROCESS *openDB (char *benutzername, char* password)
{
   printf("\nDer Benutzername ist %s\nDas Password ist %s\n",
           benutzername,password);
}

struct logrec *getLogin (char *benutzername, char *password)
{
    struct logrec *loginstruct;
    WINDOW *win;

    initscr();

    loginstruct = malloc(sizeof (struct logrec));
    win = newwin(4,45,10,15);
    box(win,'*','*');
    if ((benutzername == NULL) || (strlen (benutzername) == 0))
    {
        benutzername = malloc(30);
        mvwaddstr (win,1,2,"Benutzername : ");
        if (wgetstr(win,benutzername) == NULL)
        {
            echo();
            endwin();
            printf("\nKein Benutzername angegeben! Abbruch!\n");
            exit(1);
        }
    }
    strcpy(loginstruct->benutzername,benutzername);
    if ((password == NULL) || (strlen (password) == 0))
    {
        noecho();
        password = malloc(30);
        mvwaddstr (win,2,2,"Password : ");
        if (wgetstr(win,password) != 1)
        {
            echo();
            endwin();
            printf("\nKein Password angegeben! Abbruch!\n");
            exit(1);
        }
    }
    strcpy(loginstruct->password,password);

    echo();
    endwin();

  printf("\n%s %s\n",loginstruct->benutzername,loginstruct->password);
    return (loginstruct);
}
```

MGETLOG
4.4.91

```
#module mgetlog

/*
**++
**  FACILITY:
**
**      get database name
**          user name
**          password
**
**  ABSTRACT:
**
**      This module is specified to help the programmer in his task
**      of specifying the database, the username and the password.
**      If the database name is specified it will not be prompted.
**      If the user name is specified it will not be promted.
**      If the password is specified it will not be prompted (this
**          case should ever occur.).
**      If any of this string is not specified it will be prompted.
**
**  AUTHORS:
**
**      Lutz-Peter Kurdelski
**
**
**  CREATION DATE:      1990-11-14
**
**  MODIFICATION HISTORY:
**
**--
*/


/*
**
**  INCLUDE FILES
**
*/

#include curses


/*
**
**  MACRO DEFINITIONS
**
*/

#define bool int
#define aStringSize 31
#define aShortStringSize 21

typedef  struct logrec {
    char username [aStringSize],
         password [aShortStringSize];
    };

typedef  struct llogrec {
    char databaseName [aStringSize],
         username [aStringSize],
         password [aShortStringSize];
    };
```

```
/*
**++
**  FUNCTIONAL DESCRIPTION:
**
**      getLogin (char *, char *);
**
**      Reads the username and the password
**
**  FORMAL PARAMETERS:
**
**      username
**      password
**
**  IMPLICIT INPUTS:
**
**
**
**  IMPLICIT OUTPUTS:
**
**
**
**  FUNCTION VALUE:
**
**      pointer to a structure containing the username and the password.
**
**  SIDE EFFECTS:
**
**
**--
*/
struct logrec *getLogin ( char * username, char * password)
{
    struct logrec * loginstruct;
    WINDOW * win;

    /* reserving the space for the container */
    loginstruct = malloc( sizeof (struct logrec));

    /* initialize the window */
    initscr();
    win = newwin(4,45,10,15);
    box(win,'|','-');

    if ((username == NULL) || (strlen (username) == 0))
    {
        username = malloc(aStringSize);
        mvwaddstr (win,1,2,"Username : ");
        if (wgetstr (win, username) == NULL)
        {
            return (NULL);
        }
    }
    strcpy(loginstruct->username,username);
    if ((password == NULL) || (strlen (password) == 0))
    {
        noecho();
        password = malloc(aShortStringSize);
        mvwaddstr (win,2,2,"Password : ");
        if (wgetstr (win, password) != 1)
        {
            return (NULL);
        }
    }
    strcpy(loginstruct->password,password);
    endwin();
    return (loginstruct);
}
```

```
/*
**++
**  FUNCTIONAL DESCRIPTION:
**
**      getLLogin (char *, char *);
**
**      Reads the database name, the username and the password
**
**  FORMAL PARAMETERS:
**
**      database name
**      username
**      password
**
**  IMPLICIT INPUTS:
**
**
**
**  IMPLICIT OUTPUTS:
**
**
**
**  FUNCTION VALUE:
**
**      pointer to a structure containing
**          the database name, the username and the password.
**
**  SIDE EFFECTS:
**
**
**--
*/
struct llogrec *getLLogin ( char * databaseName,
                            char * username,
                            char * password)
{
    struct llogrec * loginstruct;
    WINDOW * win;

    /* reserving the space for the container */
    loginstruct = malloc( sizeof (struct llogrec));

    /* initialize the window */
    initscr();
    win = newwin(5,45,10,15);
    box(win,'|','-');

    if ((databaseName == NULL) || (strlen (databaseName) == 0))
    {
        databaseName = malloc(aStringSize);
        mvwaddstr (win,1,2,"Database : ");
        if (wgetstr (win, databaseName) == NULL)
        {
            return (NULL);
        }
    }
    else
    {
        mvwaddstr (win,1,2,"DATABASE : ");
        mvwaddstr (win,1,13,databaseName);
    }
    strcpy(loginstruct->databaseName,databaseName);
    if ((username == NULL) || (strlen (username) == 0))
    {
        username = malloc(aStringSize);
        mvwaddstr (win,2,2,"Username : ");
        if (wgetstr (win, username) == NULL)
        {
            return (NULL);
        }
    }
    else
    {
        mvwaddstr (win,2,2,"USERNAME : ");
        mvwaddstr (win,2,13,username);
    }
    strcpy(loginstruct->username,username);
    if ((password == NULL) || (strlen (password) == 0))
    {
        noecho();
        password = malloc(aShortStringSize);
        mvwaddstr (win,3,2,"Password : ");
        if (wgetstr (win, password) != 1)
        {
            return (NULL);
        }
    }
    strcpy(loginstruct->password,password);
    endwin();

    return (loginstruct);
}
```

ASK.FOR_PW

2.10.90

```
C.........................................................................
C
C       Unterprogramm "ask_for_pw" fragt Datenbank-Password ab, ohne dass
C       das Echo auf dem Bildschirm erscheint. Die Character-Variable
C       "password" hat eine Laenge von 20 Bytes, sie muss deshalb im
C       rufenden Programm ebenfalls als character*20 definiert werden.
C       Das Unterprogramm nutzt DEC-Routinen zur Bildschirmsteuerung, die
C       mit drei Include-Befehlen bekanntgemacht werden.
C       Grundversion von M. Reinke.
C       Aenderungen am 2.10.90 von A. Maul
C
C.........................................................................

        subroutine ask_for_pw(password)

C       Routinen zur Bildschirmsteuerung

        include '($smgdef)'
        include '($ttdef)'
        include '($tt2def)'

C       Laengendefinition des Passwords

        character*20 password

C       Echo ausschalten

        call smg$create_pasteboard(ipb)
        no_echo=tt$m_noecho
        call smg$set_term_characteristics(ipb,no_echo)

C       Password abfragen, $ belaesst Cursor in der gleichen Zeile

        print '('' Password: ''$)'
        read(5,'(a)') password

C       Echo wieder einschalten

        call smg$set_term_characteristics(ipb,,,no_echo)
        print '(1x)'

        return
        end
```