

```

subroutine prim
implicit none

integer n, init
c
c dos
c   n = init(8300000,2147483647,'prim.dat')
c vms
c   n = init(830000,2147483647,'prim.dat')
if (n .eq. 0) then
    call menue
else
    Write(*,*)'Fehler bei Init; ErrCode = ',n
endif

end

c-----init
c Declaration von globalen Variablen
c max_int = größter Integer. Bis maximal hier können Primzahlen berechnet
werden
c maxAnzahlPrim = Maximale Anzahl der zu speichernden Primzahlen
c berechneteP = Anzahl der berechneten Primzahlen
c datei = Arbeitsdatei (enthält die Primzahlen)

c function init(lmaxAnzahlPrim,lmax_int,ldatei) = errCode
c   errCode n = 0 => fehlerfreier Ablauf
c   -1 => I/O Fehler beim Öffnen der Datei
c   -3 => I/O Schreibfehler
c-----

function init(lmaxAnzahlPrim, lmax_int, ldatei)
implicit none

integer lmaxAnzahlPrim, lmax_int, init
character*(*) ldatei

c ***global***
integer maxAnzahlPrim, max_int, bereP, bisInt
character*12 datei
common /pri/ maxAnzahlPrim, max_int,bereP, datei, bisInt

integer P(2), n

max_int=lmax_int
datei=ldatei
maxAnzahlPrim=lmaxAnzahlPrim

c ***Festlegung der ersten 2 Primzahlen
P(1)=2
P(2)=3
bereP = 2
bisInt = 3
open(10,FILE=datei,STATUS='UNKNOWN',ACCESS='DIRECT',
.FORM='UNFORMATTED',RECL=4,ERR=9001)
write(10,REC=1,ERR=9003)p(1)
write(10,REC=2,ERR=9003)p(2)
close(10)
n = 0
goto 9999

9001 n = -1
goto 9999

9003 n = -3
close(10)
goto 9999

9999 init = n

end

```

```

c-----write_file
c function write_file(columns, seperator, file, typ) = errCode
c
c   columns = bei Formatted IO. Anzahl der Spalten
c   seperator = Trennzeichen zwischen den Datensätzen
c   file = Dateiname der zu erzeugenden Datei
c   typ
c       1 = Sequential, unformatted
c       2 = Sequential, formatted
c       3 = Direct(random Access), unformatted
c       4 = Direct(random Access), formatted
c
c errCode n = 0 => fehlerfreier Ablauf
c   -1 => I/O Fehler beim Öffnen der internen Datei
c   -2 => I/O Lesefehler der internen Datei
c   -5 => I/O Datei file existiert schon oder kann
c         nicht erzeugt werden
c   -6 => I/O Schreibfehler der Datei file
c   -7 => falscher Typ angegeben
c   -8 => Angabe columns fehlt oder falsch
c-----

function write_file(columns, seperator, file, typ)
implicit none

integer write_file, columns, typ
character seperator*1, file*12

c ***global***
integer maxAnzahlPrim, max_int, bereP, bisInt
character*12 datei
common /pri/ maxAnzahlPrim, max_int, bereP, datei, bisInt

integer P(bereP), i, j, k, l, n
character form*30
character*10 date, time

if (typ .lt. 1 .or. typ .gt. 4) goto 9007

goto (201,202,203,204)typ
201 open(20,FILE=file,STATUS='new',ACCESS='SEQUENTIAL',
.FORM='UNFORMATTED',ERR=9005)
goto 299
202 open(20,FILE=file,STATUS='new',ACCESS='SEQUENTIAL',
.FORM='FORMATTED',ERR=9005)
goto 299
203 open(20,FILE=file,STATUS='new',ACCESS='DIRECT',
.FORM='UNFORMATTED',RECL=4,ERR=9005)
goto 299
204 open(20,FILE=file,STATUS='new',ACCESS='DIRECT',
.FORM='FORMATTED',RECL=11,ERR=9005)
goto 299

299 open(10,FILE=datei,STATUS='UNKNOWN',ACCESS='DIRECT',
.FORM='UNFORMATTED',RECL=4,ERR=9001)
do i=1,bereP
    read(10,REC=i,ERR=9002)j
    P(i)=j
enddo
close(10)

c ***sequential formatted***Header
if (typ .eq. 2) then
4002 Format('Berechnete Primzahlen von 2 bis : ',I10)
4003 Format(21(' '), ' erste Primzahl : ',I10)
4004 Format(21X, ' letzte Primzahl : ',I10)
4005 Format(21X, 'Anzahl Primzahlen : ',I10)

call date_and_Time(date,time)

```

```

write(20,*)'Filename :',file
write(20,*)'Date      :',date
write(20,*)'Time      :',time
write(20,4002)bisInt
write(20,4003)2
write(20,4004)P(bereP)
write(20,4005)bereP
endif

j = 0
if (mod(typ,2) .eq. 0)then
  if(typ .eq. 2) then
    if (columns .lt. 1) goto 9008
    write(form,'(I10)')columns
    form = ' '//form(1:10)//'(I10, '//seperator//''')'
    write(*,*)form,'columns : ',columns
    call press_key
    do i=1,bereP/columns
      write(20,form,ERR=9006)(p(j*columns+k),k=1,columns)
      j = j + 1
    enddo
    l = mod(bereP,columns)
    write(20,form,ERR=9006)(p(j*columns+k),k=1,l)
    write(20,*)'-1'
  else
    form = '(I10, '//seperator//''')'
    do i=1,bereP
      write(20,form,REC=i,ERR=9006)p(i)
    enddo
  endif
endif
else
  do i=1,bereP
    if(typ .eq. 1) then
      write(20,Err=9006)p(i)
    else
      write(20,REC=i,ERR=9006)p(i)
    endif
  enddo
endif

close(20)
close(10)
n = 0
goto 9999

9001 n = -1
close(20)
goto 9999

9002 n = -2
close(10)
close(20)
goto 9999

9005 n = -5
goto 9999

9006 n = -6
close(20)
goto 9999

9007 n = -7
goto 9999

9008 n = -8
close(20)
goto 9999

9999 write_file = n

end

```

```

c-----primCalculator
c function primCalculator(letztePrim) = errCode
c letztePrim enthält die letzte gefundene Primzahl als Rückgabeparameter
c
c errCode n = 0 => fehlerfreier Ablauf
c          -1 => I/O Fehler beim Öffnen der Datei
c          -2 => I/O Lesefehler
c          -3 => I/O Schreibfehler
c          -4 => interner Buffer zu klein. Kann Prim nicht speichern
c-----

function primCalculator(letztePrim)
implicit none

integer primCalculator, letztePrim

c ***global***
integer maxAnzahlPrim, max_int, bereP, bisInt
character*12 datei
common /pri/ maxAnzahlPrim, max_int, bereP, datei, bisInt

integer P(maxAnzahlPrim)
integer i, j, rest, max_j, n

write(*,*)'naja wohl ein test oder was'
open(10,FILE=datei,STATUS='UNKNOWN',ACCESS='DIRECT',
.FORM='UNFORMATTED',RECL=4,ERR=9001)
do i=1,bereP
  read(10,REC=i,ERR=9002)j
  P(i)=j
enddo

do i=P(bereP)+2,bisInt,2
  j = 1
  max_j = ifix(sqrt(float(i)))

10  rest = mod(i, P(j))
    if (rest .eq. 0) goto 30
    j = j + 1
    if ((P(j) .le. max_j) .and. (j .le. bereP)) goto 10

c    ***ist Primzahl
    bereP = bereP + 1
    P(bereP) = i
    letztePrim = i
    write(10,REC=bereP,ERR=9003)i
    if (bereP .ge. maxAnzahlPrim) goto 9004
    write(*,*)i

30  enddo
    call press_key
    close(10)
    n = 0
    goto 9999

9001 n = -1
goto 9999

9002 n = -2
close(10)
goto 9999

9003 n = -3
close(10)
goto 9999

9004 n = -4
close(10)
goto 9999

```

```

prim.for                                03.01.2002
9999 primCalculater = n
      end

c-----showPrimIntervall
c function showPrimIntervall (von, bis) = errCode
c
c errCode n = 0 => fehlerfreier Ablauf
c          -1 => I/O Fehler beim Öffnen der Datei
c          -2 => I/O Lesefehler
c          -3 => ungültiger Intervall
c-----
      function showPrimIntervall (von, bis)
      implicit none

      integer showPrimIntervall, von, bis

c      ***global***
      integer maxAnzahlPrim, max_int, bereP, bisInt
      character*12 datei
      common /pri/ maxAnzahlPrim, max_int, bereP, datei, bisInt

      integer P(5), i, j, k, n

      if ((von .lt. 2).or.(bis .gt. bisInt).or.(von .gt. bis))
      .goto 9003

      open(10,FILE=datei,STATUS='UNKNOWN',ACCESS='DIRECT',
      .FORM='UNFORMATTED',RECL=4,ERR=9001)
      i = 0
10     i = i + 1
      read(10,REC=i,ERR=9002)j
      if ((j .lt. von).and. (i .lt. bereP)) goto 10

20     k = 0
      k = k + 1
      P(mod(k-1,5)+1)=j
      if (j .gt. bis) goto 30
      if (mod(k,5) .eq. 0) then
      write(*,4000)k,(P(j),j=1,5)
      endif
      if ((j .eq. bis).or.(i .eq. bereP)) goto 30
      i = i + 1
      read(10,REC=i,ERR=9002)j
      goto 20

4000 Format(1X,'Anzahl',I10,'|',5I10)

30     if ((j .ne. bis).and.(i .ne. bereP)) k = k - 1
      write(*,4000)k,(P(j),j=1,mod(k,5))

      call press_key
      n = 0
      close(10)
      goto 9999

9001 n = -1
      goto 9999

9002 n = -2
      close(10)
      goto 9999

9003 n = -3
      goto 9999

9999 showPrimIntervall = n

      end

```

```

prim.for                                03.01.2002

c-----showPrimK
c function showPrimK(i) = errCode
c
c errCode n = 0 => fehlerfreier Ablauf
c          -1 => I/O Fehler beim Öffnen der Datei
c          -2 => I/O Lesefehler
c          -3 => ungültiges k
c-----
      function showPrimK(k)
      implicit none

      integer showPrimk, k

c      ***global***
      integer maxAnzahlPrim, max_int, bereP, bisInt
      character*12 datei
      common /pri/ maxAnzahlPrim, max_int, bereP, datei, bisInt

      integer j,n

      if ((k .lt. 1).or.(k .gt. bereP)) goto 9003

      open(10,FILE=datei,STATUS='UNKNOWN',ACCESS='DIRECT',
      .FORM='UNFORMATTED',RECL=4,ERR=9001)
      read(10,REC=k,ERR=9002)j

4000 Format(1X,'Position ',I10,'|',I10)

      write(*,4000)k,j
      call press_key
      n = 0
      close(10)
      goto 9999

9001 n = -1
      goto 9999

9002 n = -2
      close(10)
      goto 9999

9003 n = -3
      close(10)
      goto 9999

9999 showPrimK = n

      end

c-----menue_header
c      PRIMZAHLEN - ERMITTLUNG
c
c      BERECHNET PRIMZAHLEN VON 2 BIS maximal : 2147483647
c      Berechnete Primzahlen von 2 bis      :      3
c                                          erste Primzahl :      2
c                                          letzte Primzahl : 4198844
c                                          Anzahl Primzahlen :      2
c
c-----
      subroutine menue_header(letztePrim)
      implicit none

      integer letztePrim

c      ***global***
      integer maxAnzahlPrim, max_int, bereP, bisInt

```

```

character*12 datei
common /pri/ maxAnzahlPrim, max_int, bereP, datei, bisInt

4000 Format(6X,'PRIMZAHLEN - ERMITTLUNG')
4001 Format(6X,'BERECHNET PRIMZAHLEN VON 2 BIS maximal : ',I10)
4002 Format(6X,'Berechnete Primzahlen von 2 bis      : ',I10)
4003 Format(27(' '), ' erste Primzahl : ',I10)
4004 Format(27X,' letzte Primzahl : ',I10)
4005 Format(6H      ,21X,'Anzahl Primzahlen : ',I10)
4006 Format(1X)
4007 Format(///)

Write(*,4007)
Write(*,4007)
Write(*,4000)
Write(*,4006)
Write(*,4001)max_int
Write(*,4002)bisInt
Write(*,4003)2
Write(*,4004)letztePrim
Write(*,4005)bereP
Write(*,4006)

end

```

```

c-----menue
c      |-----Menue-----|
c      | [1] um Primzahlen aus einer Datei einzulesen |
c      | [2] um Primzahlen in einer Datei zu schreiben |
c      | [3] um Primzahlen bis zu einem bestimmten Wert zu berechnen |
c      | [4] um Primzahlen auszugeben |
c      | [5] um das Programm zu beenden |
c      |-----|
c
c
c
cAuswahl :
c-----
c      subroutine menue
c      implicit none
c
c      ***global***
c      integer maxAnzahlPrim, max_int, bereP, bisInt
c      character*12 datei
c      common /pri/ maxAnzahlPrim, max_int, bereP, datei, bisInt
c
c      integer i, j, letztePrim, n, columns
c      integer showPrimIntervall, primCalculater, showPrimK, write_file
c      character menuetxt*60
c      character seperator*1, file*12
c
4007 Format(///)
4000 Format(6X,'|-',28('-'),'Menue',27('-'),'|-|')
4001 Format(6X,'| |','|',I1,'| ',A56,' |')
4002 Format(6X,'|-',60('-'),'|-|')
5000 Format(1X,'Auswahl :')

c      Letzte Primzahl die beim Init gesetzt wurde
c      letztePrim = bisInt

10  continue
c      call menue_header(letztePrim)

Write(*,4000)
menuetxt = 'um Primzahlen in einer Datei zu schreiben'
Write(*,4001)1,menuetxt
menuetxt =
.'um Primzahlen bis zu einem bestimmten Wert zu berechnen'
Write(*,4001)2,menuetxt

```

```

menuetxt = 'um Primzahlen auszugeben'
Write(*,4001)3,menuetxt
menuetxt = 'um das Programm zu beenden'
Write(*,4001)4,menuetxt
Write(*,4002)
Write(*,4007)
Write(*,5000)
read(*,*)i

if (i .le. 0 .or. i .gt. 4) goto 10
goto(100,200,300,400)i

100  call menue_header(letztePrim)
Write(*,4000)
menuetxt = 'Sequential, unformatted'
Write(*,4001)1,menuetxt
menuetxt = 'Sequential, formatted'
Write(*,4001)2,menuetxt
menuetxt = 'Direct(random Access), unformatted'
Write(*,4001)3,menuetxt
menuetxt = 'Direct(random Access), formatted'
Write(*,4001)4,menuetxt
menuetxt = 'um zum Hauptmenue zurueckzukehren'
Write(*,4001)5,menuetxt
Write(*,4002)
Write(*,4007)
Write(*,5000)
read(*,*)i

if (i .le. 0 .or. i .gt. 5) goto 200
if (i .eq. 5) goto 10

if (mod(i,2) .eq. 0) then
    if (i .eq. 2) then
        write(*,*)'columns :'
        read(*,*)columns
    endif
    write(*,*)'seperator[1] :'
    read(*, '(A1)')seperator
endif
write(*,*)'filename 8.3:'
read(*, '(A12)')file
write(*,*)'file :',file
n = write_file(columns, seperator, file,i)
if (n .ne. 0) then
    write(*,*)'n= ',n
    call press_key
endif

goto 100

200  continue
write(*,*)
.'Bis zu welcher Zahl soll gerechnet werden : '
read(*,*)i
bisInt = max(i,bisInt)
n = primCalculater(letztePrim)

if (n .eq. 0) goto 10
Write(*,*)'Fehler in Primzahlenberechnung ErrCode = ',n
Write(*,*)'bei Primzahl = ',letztePrim
call press_key

if (n .eq. -4) goto 10
goto 400

300  continue
call menue_header(letztePrim)
Write(*,4000)
menuetxt = 'um Primzahlen in einem Intervall auszugeben'
Write(*,4001)1,menuetxt
menuetxt = 'um die k-te Primzahl von 2 beginnt auszugeben'
Write(*,4001)2,menuetxt

```

prim.for

03.01.2002

```
    menuetxt = 'um zum Hauptmenue zurueckzukehren'
    Write(*,4001)3,menuetxt
    Write(*,4002)
    Write(*,4007)
    Write(*,5000)
    read (*,*)i

    if (i .le. 0 .or. i .gt. 3) goto 300
    goto (301,302,10)i

301  continue
     write(*,*)
     .'Geben Sie bitte den Intervall an Syntax: X,Y [2,','bisInt,']'
     read (*,*)i,j
     n = showPrimIntervall(i,j)
     if (n .ne. 0) then
         write(*,*)'Fehler bei Ausgabe; ErrCode = ',n
         call press_key
     endif
     goto 300

302  continue
     write(*,*)'Geben Sie bitte k ein : '
     read (*,*)i
     n = showPrimK(i)
     if (n .ne. 0) then
         write(*,*)'Fehler bei Ausgabe; ErrCode = ',n
         call press_key
     endif
     goto 300

400  open(10,FILE=datei,STATUS='UNKNOWN')
     close(10,STATUS='DELETE')
     write(*,*)' ende'

     end

c-----press_key
subroutine press_key()
implicit none

character*10 x

c   DOS
c   print *, ""
c   call system("PAUSE")

c   VMS
write(*,*) 'Press Enter to continue'
read(*,'(A)') x
end
```