```
! ================================================================================
! Program: etrs89_to_utm32
! Description: Converts geographic coordinates (Latitude, Longitude in ETRS89/WGS84)
!              to UTM Zone 32N coordinates (Easting, Northing).
!
! Compilation:
!   gfortran -O2 -o etrs89_to_utm32 etrs89_to_utm32.f90
!
! Usage:
!   ./etrs89_to_utm32 48.173333 11.635
! ================================================================================

program etrs89_to_utm32
   implicit none

   ! Praezisionsdefinition (Double Precision: 15 Stellen, 8 Bytes)
   integer, parameter :: dp = selected_real_kind(15, 307)

   ! Ellipsoidparameter (GRS80 / WGS84)
   real(dp), parameter :: pi = 3.14159265358979323846264338327950288_dp
   real(dp), parameter :: a = 6378137.0_dp               ! GroÃŸe Halbachse
   real(dp), parameter :: f = 1.0_dp/298.257222101_dp  ! Abplattung
   real(dp), parameter :: e2 = f*(2.0_dp - f)            ! 1. Exzentrizitaet quadrat
   real(dp), parameter :: k0 = 0.9996_dp                 ! MaÃŸstabsfaktor Meridian

   character(len=100) :: arg1, arg2
   real(dp) :: lat_deg, lon_deg, easting, northing
   integer :: status

   if (command_argument_count() < 2) then
      print *, 'Fehler: Zu wenige Argumente.'
      print *, 'Verwendung: ./etrs89_to_utm32 <Breitengrad> <Laengegrad>'
      print *, 'Beispiel:   ./etrs89_to_utm32 48.173333 11.635'
      stop 1
   end if

   call get_command_argument(1, arg1)
   call get_command_argument(2, arg2)

   read (arg1, *, iostat=status) lat_deg
   if (status /= 0) stop 'Fehler beim Lesen des Breitengrades'

   read (arg2, *, iostat=status) lon_deg
   if (status /= 0) stop 'Fehler beim Lesen des Laengegrades'

```

```
   call latlon_to_utm32(lat_deg, lon_deg, easting, northing)

   print '(A)', '------------------------------------------------'
   print '(A)', 'ETRS89 / WGS84 zu UTM Zone 32N Konvertierung'
   print '(A)', '------------------------------------------------'
   print '(A, F12.7, A)', 'Eingabe Breitengrad (Lat): ', lat_deg, ' deg'
   print '(A, F12.7, A)', 'Eingabe Laengegrad  (Lon): ', lon_deg, ' deg'
   print '(A)', '------------------------------------------------'
   print '(A, F12.3, A)', 'UTM Zone 32 Ostwert (Easting)  : ', easting, ' m'
   print '(A, F12.3, A)', 'UTM Zone 32 Nordwert (Northing): ', northing, ' m'
   print '(A, F10.3, F12.3)', 'UTM Ref (32N)                  : 32N ', easting, northing
   print '(A)', '------------------------------------------------'

contains

   subroutine latlon_to_utm32(lat, lon, east, north)
      implicit none
      real(dp), intent(in)  :: lat, lon
      real(dp), intent(out) :: east, north

      real(dp) :: phi, lambda, lambda0
      real(dp) :: n_val, A_coeff, B_coeff, C_coeff, D_coeff, E_coeff, S
      real(dp) :: ep2, N, T, C_val, A_val

      ! Umrechnung in Radian
      phi = lat*pi/180.0_dp
      lambda = lon*pi/180.0_dp
      lambda0 = 9.0_dp*pi/180.0_dp  ! Hauptmeridian Zone 32 (9Â° E)

      ! MeridianbogenlÃ¤nge (S)
      n_val = f/(2.0_dp - f)
      A_coeff = a*(1.0_dp + n_val**2/4.0_dp + n_val**4/64.0_dp)
      B_coeff = (3.0_dp/2.0_dp)*n_val*(1.0_dp - n_val**2/8.0_dp)
      C_coeff = (15.0_dp/16.0_dp)*(n_val**2)*(1.0_dp - n_val**2/4.0_dp)
      D_coeff = (35.0_dp/48.0_dp)*(n_val**3)
      E_coeff = (315.0_dp/512.0_dp)*(n_val**4)

      S = A_coeff*(phi - B_coeff*sin(2.0_dp*phi) + C_coeff*sin(4.0_dp*phi) &
                   - D_coeff*sin(6.0_dp*phi) + E_coeff*sin(8.0_dp*phi))

      ! Geometrische Hilfskonstanten
      ep2 = e2/(1.0_dp - e2)
      N = a/sqrt(1.0_dp - e2*sin(phi)**2)
      T = tan(phi)**2
      C_val = ep2*cos(phi)**2
      A_val = (lambda - lambda0)*cos(phi)
```

```
 92
 93       ! Ostwert (Easting)
 94       east = 500000.0_dp + k0*N*(A_val + (1.0_dp - T + C_val)*(A_val**3)/6.0_dp &
 95                                  + (5.0_dp - 18.0_dp*T + T**2 + 72.0_dp*C_val - 58.0_dp*ep2)*(A_val**5)/120.0_dp)
 96
 97       ! Nordwert (Northing)
 98       north = k0*(S + N*tan(phi)*((A_val**2)/2.0_dp &
 99                                   + (5.0_dp - T + 9.0_dp*C_val + 4.0_dp*(C_val**2))*(A_val**4)/24.0_dp &
100                                   + (61.0_dp - 58.0_dp*T + T**2 + 600.0_dp*C_val - 330.0_dp*ep2)*(A_val**6)/720.0_dp))
101
102    end subroutine latlon_to_utm32
103
104 end program etrs89_to_utm32
```