```fortran
! 3456789 123456789 223456789 323456789 423456789 523456789 623456789 723456789 823456789 923456789 023456789 123456789 223456789 32
program least_squares

    use, intrinsic :: iso_fortran_env,  only : REAL64, compiler_options, compiler_version

    implicit none

    character ( len = * ), parameter :: fmt_datecom = '( /, " - - - - - - - - -", //, I5, 2 ( "-", I2.2 ), I3, 2 ( ":", I2.2 ), / )'

    integer, parameter :: rp = selected_real_kind ( REAL64 ),  m = 9, n = 2

    ! rank one
    real ( kind = rp ), parameter :: one ( 1 : m ) = [ 1.0_rp, 1.0_rp, 1.0_rp, 1.0_rp, 1.0_rp, 1.0_rp, 1.0_rp, 1.0_rp, 1.0_rp ]
    real ( kind = rp ), parameter :: x   ( 1 : m ) = [ 1.0_rp, 2.0_rp, 3.0_rp, 4.0_rp, 5.0_rp, 6.0_rp, 7.0_rp, 8.0_rp, 9.0_rp ]
    real ( kind = rp ), parameter :: T   ( 1 : m ) = [ 15.6_rp, 17.5_rp, 36.6_rp, 43.8_rp, 58.2_rp, 61.6_rp, 64.2_rp, 70.4_rp, &
                                                       98.8_rp ]

    real ( kind = rp ) :: residual ( 1 : m ) = 0.0_rp
    real ( kind = rp ) :: sigma    ( 1 : n ) = 0.0_rp

    ! scalars
    real ( kind = rp ) :: a = 0.0_rp, b = 0.0_rp, c = 0.0_rp, d = 0.0_rp, e = 0.0_rp, det = 0.0_rp, a0 = 0.0_rp, a1 = 0.0_rp
    real ( kind = rp ) :: t2 = 0.0_rp, psd = 0.0_rp

    ! containers for date and time
    integer :: dt_values ( 1 : 8 ) = 0

            ! inner products
            a = dot_product ( one, one )
            b = dot_product ( one, x )
            c = dot_product ( x,   x )
            d = dot_product ( one, T )
            e = dot_product ( x,   T )
            ! determinant of Gram matrix
            det = a * c - b**2

            ! equations 10.26
            a0 = ( d * c - b * e ) / det
            a1 = ( a * e - b * d ) / det

            residual = a0 * one + a1 * x - T
            t2 = dot_product ( residual, residual )
            ! equation 6.19
            psd = t2 / real ( m - n, kind = rp )
            ! equations 6-21, 22
            sigma = sqrt( psd * [ c, a ] / det )

            ! compare to values in table 6-1, p. 93
            write ( * , * ) "particular least squares solution"
            write ( * , * ) "intercept a0 = ", a0, " +/- ", sigma( 1 )
            write ( * , * ) "slope     a1 = ", a1, " +/- ", sigma( 2 )

        ! execution complete - tag output
        call date_and_time ( VALUES = dt_values )
            write ( * , fmt_datecom ) dt_values ( 1 : 3 ), dt_values ( 5 : 7 )

        write ( * , '( "compiler version: ", g0, "."    )' ) compiler_version ( )
        write ( * , '( "compiler options: ", g0, ".", / )' ) compiler_options ( )

        stop 'Successful termination for "least_squares_plus.f08".'

end program least_squares

! dantopa@Xiuhcoatl.local:least-squares $ pwd
! /Volumes/repos/github/jop/fortran/genesis/least-squares

! dantopa@Xiuhcoatl.local:least-squares $ rm a.*

! dantopa@Xiuhcoatl.local:least-squares $ gfortran least_squares_plus.f08

! dantopa@Xiuhcoatl.local:least-squares $ ./a.out
!  particular least squares solution
!  intercept a0 =    4.8138888888888891       +/-    4.8862063121833534
!  slope     a1 =    9.4083333333333332       +/-   0.86830164765636075
!
!  - - - - - - - - -
!
!  2022-12-05 21:52:39
!
! compiler version: GCC version 12.2.0.
! compiler options: -fPIC -mmacosx-version-min=13.0.0 -mtune=core2.
!
! STOP Successful termination for "least_squares_plus.f08".
```