```
!  This is a test program for version 1.4 of module fm_double_int, which contains the interface
!  routines allowing quadruple-precision real variables in the user's program to be used in
!  assignments, arithmetic, and comparisons involving type (fm), (im), and (zm) variables.
!  The same operations are provided as those in the basic module fmzm for single or double
!  precision variables.

!  All of the routines in module fm_double_int are tested, and if all tests are completed
!  successfully, this line is printed:

!  280 cases tested.  No errors were found.


      module test_vars

      use fmvals
      use fmzm
      use fm_double_int

      type (fm), save ::  m_a, mfm1, mfm2, mfm3, mfm4, mfm6,  &
                          mfmv1(3),  mfmv2(3),                &
                          mfma(3, 3), mfmb(3, 3)

      type (im), save :: m_j,  mim1,  mim2,  mim3, mim4, mim5
      type (im), save, dimension(3)    :: mimv1, mimv2
      type (im), save, dimension(3, 3) :: mima2, mimb2

      type (zm), save :: m_z, mzm1, mzm2, mzm3, mzm4, mzm5,  &
                         mzmv1(3),    mzmv2(3),               &
                         mzma2(3, 3), mzmb2(3, 3)

      integer (double_int), save :: di1, di2, di3, di4, di5, div(3), div2(3, 3)
      real, save :: rv(3), rv2(3, 3)
      double precision, save :: dv(3), dv2(3, 3)
      complex, save :: cv(3), cv2(3, 3)
      complex (kind(0.0d0)), save :: cdv(3), cdv2(3, 3)

      integer, save :: klog, kwsave, ncase, nerror
      real, save :: time1, time2

      end module test_vars

      module test_a
      use test_vars

      contains

      subroutine test1

!           Test the = assignment interface.

      implicit none

      write (kw, "(/' Testing the derived type = interface.')")

      ncase = 1
      di4 = mfm1
      if (di4 /= 581) call prterr(kw)
```

```
ncase = 2
di4 = mim1
if (di4 /= 661) call prterr(kw)

ncase = 3
di4 = mzm1
if (di4 /= 731) call prterr(kw)

ncase = 4
mfm3 = di2
call fm_st2m('1234567890123', mfm4)
call fm_sub(mfm3, mfm4, mfm6)
call fm_eq(mfm6, mfm4)
call fm_abs(mfm4, mfm6)
call fm_eq(mfm6, mfm4)
call fm_st2m('0', mfm3)
if (fm_compare(mfm4, 'gt', mfm3)) call prterr(kw)

ncase = 5
mfm3 = di2
call fm_st2m('1234567890123', mfm4)
call fm_sub(mfm3, mfm4, mfm6)
call fm_equ(mfm6, mfm4, ndig, ndig)
call fm_abs(mfm4, mfm6)
call fm_equ(mfm6, mfm4, ndig, ndig)
call fm_st2m('0', mfm3)
if (fm_compare(mfm4, 'gt', mfm3)) call prterr(kw)

ncase = 6
mfm3 = di2
call fm_st2m('1234567890123', mfm4)
call fm_sub_r2(mfm3, mfm4)
call fm_abs(mfm4, mfm6)
call fm_equ(mfm6, mfm4, ndig, ndig)
call fm_st2m('0', mfm3)
if (fm_compare(mfm4, 'gt', mfm3)) call prterr(kw)

ncase = 7
mfm3 = di2
mfm4 = to_double_int(mfm3)
call fm_sub_r2(mfm3, mfm4)
call fm_abs(mfm4, mfm6)
call fm_equ(mfm6, mfm4, ndig, ndig)
call fm_st2m('0', mfm3)
if (fm_compare(mfm4, 'gt', mfm3)) call prterr(kw)

ncase = 8
mfm3 = di2
call fm_st2m('1234567890123', mfm4)
call fm_equ(mfm3, mfm6, ndig, ndig)
call fm_sub_r1(mfm6, mfm4)
call fm_equ(mfm6, mfm4, ndig, ndig)
call fm_abs(mfm4, mfm6)
call fm_equ_r1(mfm6, ndig, ndig)
call fm_eq(mfm6, mfm4)
call fm_st2m('0', mfm3)
if (fm_compare(mfm4, 'gt', mfm3)) call prterr(kw)
```

```
ncase = 9
mim3 = di2
call im_st2m('1234567890123', mim4)
call im_sub(mim3, mim4, mim5)
call im_eq(mim5, mim4)
call im_st2m('0', mim3)
if (im_compare(mim4, 'gt', mim3)) call prterr(kw)

ncase = 10
mim3 = to_im(di2)
call im_st2m('1234567890123', mim4)
call im_sub(mim3, mim4, mim5)
call im_eq(mim5, mim4)
call im_st2m('0', mim3)
if (im_compare(mim4, 'gt', mim3)) call prterr(kw)

ncase = 11
mim3 = to_im(di2)
mim4 = to_double_int(mim3)
call im_sub(mim3, mim4, mim5)
call im_eq(mim5, mim4)
call im_st2m('0', mim3)
if (im_compare(mim4, 'gt', mim3)) call prterr(kw)

ncase = 12
mzm3 = di2
call zm_st2m('1234567890123', mzm4)
call zm_sub(mzm3, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
call zm_abs(mzm4, mfm4)
call fm_st2m('0', mfm3)
if (fm_comp(mfm4, 'gt', mfm3)) call prterr(kw)

ncase = 13
mzm3 = to_zm(di2)
call zm_st2m('1234567890123', mzm4)
call zm_sub(mzm3, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
call zm_abs(mzm4, mfm4)
call fm_st2m('0', mfm3)
if (fm_comp(mfm4, 'gt', mfm3)) call prterr(kw)

ncase = 14
mzm3 = to_zm(di2)
mzm4 = to_double_int(mzm3)
call zm_sub(mzm3, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
call zm_abs(mzm4, mfm4)
call fm_st2m('0', mfm3)
if (fm_comp(mfm4, 'gt', mfm3)) call prterr(kw)

ncase = 15
di1 = 123
mzm3 = to_zm(di1, di2)
call zm_st2m('123 + 1234567890123 i', mzm4)
call zm_sub(mzm3, mzm4, mzm5)
call zm_eq(mzm5, mzm4)
```

```fortran
      call zm_abs(mzm4, mfm4)
      call fm_st2m('0', mfm3)
      if (fm_comp(mfm4, 'gt', mfm3)) call prterr(kw)

      end subroutine test1

      subroutine test2

!   Test the derived type == interface.

      implicit none

      write (kw, "(/' Testing the derived type == interface.')")

      ncase = 16
      di1 = 123
      m_a = di1
      if (.not.(m_a == di1)) then
          call errprt_fm('  ==  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
      endif

      ncase = 17
      di1 = 123
      m_a = di1
      if (.not.(di1 == m_a)) then
          call errprt_fm('  ==  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
      endif

      ncase = 18
      di1 = 123
      m_j = di1
      if (.not.(m_j == di1)) then
          call errprt_im('  ==  ', m_j, 'm_j', m_j, 'm_j')
      endif

      ncase = 19
      di1 = 123
      m_j = di1
      if (.not.(di1 == m_j)) then
          call errprt_im('  ==  ', m_j, 'm_j', m_j, 'm_j')
      endif

      ncase = 20
      di1 = 123
      m_z = di1
      if (.not.(m_z == di1)) then
          call errprt_zm('  ==  ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
      endif

      ncase = 21
      di1 = 123
      m_z = ( 123.0 , 34.5 )
      if (m_z == di1) then
          call errprt_zm('  ==  ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
      endif

      ncase = 22
      di1 = 123
```

```fortran
      m_z = di1
      if (.not.(di1 == m_z)) then
          call errprt_zm('  ==  ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
      endif

      ncase = 23
      di1 = 123
      m_z = ( 123.0 , 34.5 )
      if (di1 == m_z) then
          call errprt_zm('  ==  ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
      endif

      return
      end subroutine test2

      subroutine test3

!  Test the derived type /= interface.

      implicit none

      write (kw, "(/' Testing the derived type /= interface.')")

      ncase = 24
      di1 = 123
      m_a = 1 + di1
      if (.not.(m_a /= di1)) then
          call errprt_fm('  /=  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
      endif

      ncase = 25
      di1 = 123
      m_a = 1 + di1
      if (.not.(di1 /= m_a)) then
          call errprt_fm('  /=  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
      endif

      ncase = 26
      di1 = 123
      m_j = 1 + di1
      if (.not.(m_j /= di1)) then
          call errprt_im('  /=  ', m_j, 'm_j', m_j, 'm_j')
      endif

      ncase = 27
      di1 = 123
      m_j = 1 + di1
      if (.not.(di1 /= m_j)) then
          call errprt_im('  /=  ', m_j, 'm_j', m_j, 'm_j')
      endif

      ncase = 28
      di1 = 123
      m_z = 1 + di1
      if (.not.(m_z /= di1)) then
          call errprt_zm('  /=  ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
      endif
```

```fortran
      ncase = 29
      di1 = 123
      m_z = ( 123.0 , 34.5 )
      if (.not.(m_z /= di1)) then
          call errprt_zm('  /=  ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
      endif


      ncase = 30
      di1 = 123
      m_z = 1 + di1
      if (.not.(di1 /= m_z)) then
          call errprt_zm('  /=  ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
      endif


      ncase = 31
      di1 = 123
      m_z = ( 123.0 , 34.5 )
      if (.not.(di1 /= m_z)) then
          call errprt_zm('  /=  ', m_z, 'm_z', m_z, 'm_z', m_z, 'm_z')
      endif


      return
      end subroutine test3


      subroutine test4

!   Test the derived type > interface.

      implicit none

      write (kw, "(/' Testing the derived type > interface.')")

      ncase = 32
      di1 = 123
      m_a = di1 + 1
      if (.not.(m_a > di1)) then
          call errprt_fm('   >  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
      endif

      ncase = 33
      di1 = 123
      m_a = di1 - 1
      if (.not.(di1 > m_a)) then
          call errprt_fm('   >  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
      endif

      ncase = 34
      di1 = 123
      m_j = di1 + 1
      if (.not.(m_j > di1)) then
          call errprt_im('   >  ', m_j, 'm_j', m_j, 'm_j')
      endif

      ncase = 35
      di1 = 123
      m_j = di1 - 1
      if (.not.(di1 > m_j)) then
          call errprt_im('   >  ', m_j, 'm_j', m_j, 'm_j')
```

```fortran
        endif

        return
        end subroutine test4

        subroutine test5

!   Test the derived type >= interface.

        implicit none

        write (kw, "(/' Testing the derived type >= interface.')")

        ncase = 36
        di1 = 123
        m_a = di1 + 1
        if (.not.(m_a >= di1)) then
            call errprt_fm('  >=  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
        endif

        ncase = 37
        di1 = 123
        m_a = di1 - 1
        if (.not.(di1 >= m_a)) then
            call errprt_fm('  >=  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
        endif

        ncase = 38
        di1 = 123
        m_j = di1 + 1
        if (.not.(m_j >= di1)) then
            call errprt_im('  >=  ', m_j, 'm_j', m_j, 'm_j')
        endif

        ncase = 39
        di1 = 123
        m_j = di1 - 1
        if (.not.(di1 >= m_j)) then
            call errprt_im('  >=  ', m_j, 'm_j', m_j, 'm_j')
        endif

        return
        end subroutine test5

        subroutine test6

!   Test the derived type < interface.

        implicit none

        write (kw, "(/' Testing the derived type < interface.')")

        ncase = 40
        di1 = 123
        m_a = di1 - 2
        if (.not.(m_a < di1)) then
            call errprt_fm('   <  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
        endif
```

```fortran
      ncase = 41
      di1 = 123
      m_a = di1 + 2
      if (.not.(di1 < m_a)) then
          call errprt_fm('    <  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
      endif

      ncase = 42
      di1 = 123
      m_j = di1 - 2
      if (.not.(m_j < di1)) then
          call errprt_im('    <  ', m_j, 'm_j', m_j, 'm_j')
      endif

      ncase = 43
      di1 = 123
      m_j = di1 + 2
      if (.not.(di1 < m_j)) then
          call errprt_im('    <  ', m_j, 'm_j', m_j, 'm_j')
      endif

      return
      end subroutine test6

      subroutine test7

!  Test the derived type <= interface.

      implicit none

      write (kw, "(/' Testing the derived type <= interface.')")

      ncase = 44
      di1 = 123
      m_a = di1 - 2
      if (.not.(m_a <= di1)) then
          call errprt_fm('   <=  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
      endif

      ncase = 45
      di1 = 123
      m_a = di1 + 2
      if (.not.(di1 <= m_a)) then
          call errprt_fm('   <=  ', m_a, 'm_a', m_a, 'm_a', m_a, 'm_a')
      endif

      ncase = 46
      di1 = 123
      m_j = di1 - 2
      if (.not.(m_j <= di1)) then
          call errprt_im('   <=  ', m_j, 'm_j', m_j, 'm_j')
      endif

      ncase = 47
      di1 = 123
      m_j = di1 + 2
      if (.not.(di1 <= m_j)) then
```

```fortran
          call errprt_im('  <=  ', m_j, 'm_j', m_j, 'm_j')
      endif

      return
      end subroutine test7

      subroutine test8

!           Test the '+' arithmetic operator.

      implicit none

      write (kw, "(/' Testing the derived type + interface.')")

      ncase = 48
      mfm3 = di2 + mfm1
      call fm_st2m('1234567890123', mfm4)
      call fm_add(mfm4, mfm1, mfm6)
      call fm_eq(mfm6, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 49
      mfm3 = di2 + mfm1
      call fm_st2m('1234567890123', mfm4)
      call fm_add_r1(mfm4, mfm1)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 50
      mfm3 = di2 + mfm1
      call fm_st2m('1234567890123', mfm4)
      call fm_add_r2(mfm1, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 51
      mim3 = di2 + mim1
      call im_st2m('1234567890123', mim4)
      call im_add(mim4, mim1, mim5)
      call im_eq(mim5, mim4)
      if (.not.(mim3 == mim4)) call prterr(kw)

      ncase = 52
      mzm3 = di2 + mzm1
      call zm_st2m('1234567890123', mzm4)
      call zm_add(mzm4, mzm1, mzm5)
      call zm_eq(mzm5, mzm4)
      if (.not.(mzm3 == mzm4)) call prterr(kw)

      ncase = 53
      mfm3 = mfm1 + di2
      call fm_st2m('1234567890123', mfm4)
      call fm_add(mfm1, mfm4, mfm6)
      call fm_eq(mfm6, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 54
      mim3 = mim1 + di2
      call im_st2m('1234567890123', mim4)
      call im_add(mim1, mim4, mim5)
```

```
        call im_eq(mim5, mim4)
        if (.not.(mim3 == mim4)) call prterr(kw)

        ncase = 55
        mzm3 = mzm1 + di2
        call zm_st2m('1234567890123', mzm4)
        call zm_add(mzm1, mzm4, mzm5)
        call zm_eq(mzm5, mzm4)
        if (.not.(mzm3 == mzm4)) call prterr(kw)

        end subroutine test8

        subroutine test9

!           Test the '-' arithmetic operator.

        implicit none

        write (kw, "(/' Testing the derived type - interface.')")

        ncase = 56
        mfm3 = di2 - mfm1
        call fm_st2m('1234567890123', mfm4)
        call fm_sub(mfm4, mfm1, mfm6)
        call fm_eq(mfm6, mfm4)
        if (.not.(mfm3 == mfm4)) call prterr(kw)

        ncase = 57
        mim3 = di2 - mim1
        call im_st2m('1234567890123', mim4)
        call im_sub(mim4, mim1, mim5)
        call im_eq(mim5, mim4)
        if (.not.(mim3 == mim4)) call prterr(kw)

        ncase = 58
        mzm3 = di2 - mzm1
        call zm_st2m('1234567890123', mzm4)
        call zm_sub(mzm4, mzm1, mzm5)
        call zm_eq(mzm5, mzm4)
        if (.not.(mzm3 == mzm4)) call prterr(kw)

        ncase = 59
        mfm3 = mfm1 - di2
        call fm_st2m('1234567890123', mfm4)
        call fm_sub(mfm1, mfm4, mfm6)
        call fm_eq(mfm6, mfm4)
        if (.not.(mfm3 == mfm4)) call prterr(kw)

        ncase = 60
        mim3 = mim1 - di2
        call im_st2m('1234567890123', mim4)
        call im_sub(mim1, mim4, mim5)
        call im_eq(mim5, mim4)
        if (.not.(mim3 == mim4)) call prterr(kw)

        ncase = 61
        mzm3 = mzm1 - di2
        call zm_st2m('1234567890123', mzm4)
```

```fortran
        call zm_sub(mzm1, mzm4, mzm5)
        call zm_eq(mzm5, mzm4)
        if (.not.(mzm3 == mzm4)) call prterr(kw)

      end subroutine test9

      end module test_a

      module test_b
      use test_vars

      contains

      subroutine test10

!            Test the '*' arithmetic operator.

      implicit none

      write (kw, "(/' Testing the derived type * interface.')")

      ncase = 62
      mfm3 = di2 * mfm1
      call fm_st2m('1234567890123', mfm4)
      call fm_mpy(mfm4, mfm1, mfm6)
      call fm_eq(mfm6, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 63
      mfm3 = di2 * mfm1
      call fm_st2m('1234567890123', mfm4)
      call fm_mpy_r1(mfm4, mfm1)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 64
      mfm3 = di2 * mfm1
      call fm_st2m('1234567890123', mfm4)
      call fm_mpy_r2(mfm1, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 65
      mfm3 = di2 * mfm1
      mfm4 = mfm1 * to_fm('1234567890123')
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 66
      mim3 = di2 * mim1
      call im_st2m('1234567890123', mim4)
      call im_mpy(mim4, mim1, mim5)
      call im_eq(mim5, mim4)
      if (.not.(mim3 == mim4)) call prterr(kw)

      ncase = 67
      mzm3 = di2 * mzm1
      call zm_st2m('1234567890123', mzm4)
      call zm_mpy(mzm4, mzm1, mzm5)
      call zm_eq(mzm5, mzm4)
      if (.not.(mzm3 == mzm4)) call prterr(kw)
```

```fortran
      ncase = 68
      mfm3 = mfm1 * di2
      call fm_st2m('1234567890123', mfm4)
      call fm_mpy(mfm1, mfm4, mfm6)
      call fm_eq(mfm6, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 69
      mim3 = mim1 * di2
      call im_st2m('1234567890123', mim4)
      call im_mpy(mim1, mim4, mim5)
      call im_eq(mim5, mim4)
      if (.not.(mim3 == mim4)) call prterr(kw)

      ncase = 70
      mzm3 = mzm1 * di2
      call zm_st2m('1234567890123', mzm4)
      call zm_mpy(mzm1, mzm4, mzm5)
      call zm_eq(mzm5, mzm4)
      if (.not.(mzm3 == mzm4)) call prterr(kw)

      end subroutine test10

      subroutine test11

!           Test the '/' arithmetic operator.

      implicit none

      write (kw, "(/' Testing the derived type / interface.')")

      ncase = 71
      mfm3 = di2 / mfm1
      call fm_st2m('1234567890123', mfm4)
      call fm_div(mfm4, mfm1, mfm6)
      call fm_eq(mfm6, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 72
      mfm3 = di2 / mfm1
      call fm_st2m('1234567890123', mfm4)
      call fm_eq(mfm1, mfm6)
      call fm_div_r2(mfm4, mfm6)
      call fm_eq(mfm6, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 73
      mim3 = di2 / mim1
      call im_st2m('1234567890123', mim4)
      call im_div(mim4, mim1, mim5)
      call im_eq(mim5, mim4)
      if (.not.(mim3 == mim4)) call prterr(kw)

      ncase = 74
      mzm3 = di2 / mzm1
      call zm_st2m('1234567890123', mzm4)
      call zm_div(mzm4, mzm1, mzm5)
```

```fortran
      call zm_eq(mzm5, mzm4)
      if (.not.(mzm3 == mzm4)) call prterr(kw)

      ncase = 75
      mfm3 = mfm1 / di2
      call fm_st2m('1234567890123', mfm4)
      call fm_div(mfm1, mfm4, mfm6)
      call fm_eq(mfm6, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 76
      mim3 = mim1 / di2
      call im_st2m('1234567890123', mim4)
      call im_div(mim1, mim4, mim5)
      call im_eq(mim5, mim4)
      if (.not.(mim3 == mim4)) call prterr(kw)

      ncase = 77
      mzm3 = mzm1 / di2
      call zm_st2m('1234567890123', mzm4)
      call zm_div(mzm1, mzm4, mzm5)
      call zm_eq(mzm5, mzm4)
      if (.not.(mzm3 == mzm4)) call prterr(kw)

      end subroutine test11

      subroutine test12

!           Test the '**' arithmetic operator.

      implicit none

      write (kw, "(/' Testing the derived type ** interface.')")

      ncase = 78
      mfm3 = di2 ** mfm1
      call fm_st2m('1234567890123', mfm4)
      call fm_power(mfm4, mfm1, mfm6)
      call fm_eq(mfm6, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 79
      di4 = 2
      mim3 = di4 ** mim1
      call im_st2m('2', mim4)
      call im_power(mim4, mim1, mim5)
      call im_eq(mim5, mim4)
      if (.not.(mim3 == mim4)) call prterr(kw)

      ncase = 80
      di4 = 23
      mzm3 = di4 ** mzm1
      mzm4 = di4
      call zm_power(mzm4, mzm1, mzm5)
      call zm_eq(mzm5, mzm4)
      if (.not.(mzm3 == mzm4)) call prterr(kw)

      ncase = 81
```

```fortran
      di4 = 2345
      mfm3 = mfm1 ** di4
      mfm4 = di4
      call fm_power(mfm1, mfm4, mfm6)
      call fm_eq(mfm6, mfm4)
      if (.not.(mfm3 == mfm4)) call prterr(kw)

      ncase = 82
      di4 = 17
      mim3 = mim1 ** di4
      call im_st2m('17', mim4)
      call im_power(mim1, mim4, mim5)
      call im_eq(mim5, mim4)
      if (.not.(mim3 == mim4)) call prterr(kw)

      ncase = 83
      di4 = 179
      mzm3 = mzm1 ** di4
      mzm4 = di4
      call zm_power(mzm1, mzm4, mzm5)
      call zm_eq(mzm5, mzm4)
      if (.not.(mzm3 == mzm4)) call prterr(kw)

      end subroutine test12

      subroutine test13

!           Test functions to_fm, to_im, to_zm, ..., to_double_int.

      implicit none
      integer :: j, k

      write (kw, "(/' Testing the derived type to_fm,  ..., to_double_int interfaces.')")

      ncase = 84
      div = (/  123,  -432,  567  /)
      mfmv1 = to_fm(div)
      mfmv2 = div
      do j = 1, 3
         if (.not.(mfmv1(j) == mfmv2(j))) then
            call prterr(kw)
            exit
         endif
      enddo

      ncase = 85
      di3 = 123
      mfmv1 = di3
      mfmv2 = 123
      do j = 1, 3
         if (.not.(mfmv1(j) == mfmv2(j))) then
            call prterr(kw)
            exit
         endif
      enddo

      ncase = 86
      div = (/  123,  -432,  567  /)
```

```fortran
      mfmv1 = to_fm(div)
      div = to_double_int(mfmv1)
      mfmv2 = div
      do j = 1, 3
         if (.not.(mfmv1(j) == mfmv2(j))) then
            call prterr(kw)
            exit
         endif
      enddo

      ncase = 87
      div2 = reshape( (/(11+3*j, j=1, 9)/) , shape = (/ 3, 3 /) )
      mfma = to_fm(div2)
      mfmb = div2
      do j = 1, 3
         do k = 1, 3
            if (.not.(mfma(j, k) == mfmb(j, k))) then
               call prterr(kw)
               exit
            endif
         enddo
      enddo

      ncase = 88
      di3 = 1234
      mfma = di3
      mfmb = 1234
      do j = 1, 3
         do k = 1, 3
            if (.not.(mfma(j, k) == mfmb(j, k))) then
               call prterr(kw)
               exit
            endif
         enddo
      enddo

      ncase = 89
      div2 = reshape( (/(11+3*j, j=1, 9)/) , shape = (/ 3, 3 /) )
      mfma = to_fm(div2)
      div2 = to_double_int(mfma)
      mfmb = div2
      do j = 1, 3
         do k = 1, 3
            if (.not.(mfma(j, k) == mfmb(j, k))) then
               call prterr(kw)
               exit
            endif
         enddo
      enddo

      ncase = 90
      div = (/  123,  -432,  567  /)
      mimv1 = to_im(div)
      mimv2 = div
      do j = 1, 3
         if (.not.(mimv1(j) == mimv2(j))) then
            call prterr(kw)
            exit
```

```fortran
      endif
   enddo

   ncase = 91
   di3 = 1234
   mimv1 = di3
   mimv2 = 1234
   do j = 1, 3
      if (.not.(mimv1(j) == mimv2(j))) then
         call prterr(kw)
         exit
      endif
   enddo

   ncase = 92
   div = (/  123,  -432,  567  /)
   mimv1 = to_im(div)
   div = to_double_int(mimv1)
   mimv2 = div
   do j = 1, 3
      if (.not.(mimv1(j) == mimv2(j))) then
         call prterr(kw)
         exit
      endif
   enddo

   ncase = 93
   div2 = reshape( (/(11+3*j, j=1, 9)/) , shape = (/ 3, 3 /) )
   mima2 = to_im(div2)
   mimb2 = div2
   do j = 1, 3
      do k = 1, 3
         if (.not.(mima2(j, k) == mimb2(j, k))) then
            call prterr(kw)
            exit
         endif
      enddo
   enddo

   ncase = 94
   di3 = 1234
   mima2 = di3
   mimb2 = 1234
   do j = 1, 3
      do k = 1, 3
         if (.not.(mima2(j, k) == mimb2(j, k))) then
            call prterr(kw)
            exit
         endif
      enddo
   enddo

   ncase = 95
   div2 = reshape( (/(11+3*j, j=1, 9)/) , shape = (/ 3, 3 /) )
   mima2 = to_im(div2)
   div2 = to_double_int(mima2)
   mimb2 = div2
   do j = 1, 3
```

```fortran
        do k = 1, 3
            if (.not.(mima2(j, k) == mimb2(j, k))) then
                call prterr(kw)
                exit
            endif
        enddo
    enddo

    ncase = 96
    div = (/  123,  -432,  567  /)
    mzmv1 = to_zm(div)
    mzmv2 = div
    do j = 1, 3
        if (abs(mzmv1(j)-mzmv2(j)) /= 0) then
            call prterr(kw)
            exit
        endif
    enddo

    ncase = 97
    di3 = 1234
    mzmv1 = di3
    mzmv2 = 1234
    do j = 1, 3
        if (abs(mzmv1(j)-mzmv2(j)) /= 0) then
            call prterr(kw)
            exit
        endif
    enddo

    ncase = 98
    div = (/  123,  -432,  567  /)
    mzmv1 = to_zm(div)
    div = to_double_int(mzmv1)
    mzmv2 = div
    do j = 1, 3
        if (abs(mzmv1(j)-mzmv2(j)) /= 0) then
            call prterr(kw)
            exit
        endif
    enddo

    ncase = 99
    div2 = reshape( (/(11+3*j, j=1, 9)/) , shape = (/ 3, 3 /) )
    mzma2 = to_zm(div2)
    mzmb2 = div2
    do j = 1, 3
        do k = 1, 3
            if (.not.(mzma2(j, k) == mzmb2(j, k))) then
                call prterr(kw)
                exit
            endif
        enddo
    enddo

    ncase = 100
    di3 = 1234
    mzma2 = di3
```

```fortran
      mzmb2 = 1234
      do j = 1, 3
         do k = 1, 3
            if (.not.(mzma2(j, k) == mzmb2(j, k))) then
               call prterr(kw)
               exit
            endif
         enddo
      enddo

      ncase = 101
      div2 = reshape( (/(11+3*j, j=1, 9)/) , shape = (/ 3, 3 /) )
      mzma2 = to_zm(div2)
      div2 = to_double_int(mzma2)
      mzmb2 = div2
      do j = 1, 3
         do k = 1, 3
            if (.not.(mzma2(j, k) == mzmb2(j, k))) then
               call prterr(kw)
               exit
            endif
         enddo
      enddo

      ncase = 102
      call fmm2di(mfm1%mfm, di3)
      if (to_int(mfm1) /= di3) call prterr(kw)

      ncase = 103
      call imm2di(mim1%mim, di3)
      if (to_int(mim1) /= di3) call prterr(kw)

      ncase = 104
      call fmm2di(mzm1%mzm(1), di3)
      if (to_int(mzm1) /= di3) call prterr(kw)

      end subroutine test13

      subroutine test14

!           Test the derived-type interface routines that are not used elsewhere in this program.

      implicit none

      ncase = 105
      di4 = mfm1
      call fmm2di(mfm1%mfm, di5)
      if (di4 /= di5) call prterr(kw)

      ncase = 106
      mim3 = mim1 / 13
      di5 = mim1
      di5 = di5 / 13
      if (.not.(mim3 == di5)) call prterr(kw)

      ncase = 107
      di4 = mod(mim1, to_im(13))
      di5 = mod(to_double_int(mim1), 13_double_int)
```

```fortran
      if (di4 /= di5) call prterr(kw)

      ncase = 108
      di4 = mim1
      call imm2di(mim1%mim, di5)
      if (di4 /= di5) call prterr(kw)

      ncase = 109
      di4 = mzm1
      call fmm2di(mzm1%mzm(1), di5)
      if (di4 /= di5) call prterr(kw)

      end subroutine test14

      end module test_b

      module test_c
      use test_vars

      contains

      subroutine test15

!   Test type (fm) array equal assignments.

      implicit none
      integer :: j

      write (kw, "(/' Testing derived-type array operations.')")

      kwsave = kw
      call fmsetvar(' kw = 22 ')
      call fmsetvar(' ntrace = 0 ')

      ncase = 110
      mfm3 = to_fm('234.56')
      div = mfm3
      di5 = 0
      do j = 1, 3
         di5 = di5 + abs(div(j) - 234)
      enddo
      call fm_st2m(' 1.0e-45 ', mfm4)
      if (.not.(to_fm(di5) <= mfm4)) then
          call prterr(kwsave)
      endif

      ncase = 111
      div = (/ 12, -34, 56 /)
      mfmv1 = div
      mfm3 = 0
      do j = 1, 3
         mfm3 = mfm3 + abs(mfmv1(j) - div(j))
      enddo
      call fm_st2m(' 1.0e-45 ', mfm4)
      if (.not.(mfm3 <= mfm4)) then
          call prterr(kwsave)
      endif
```

```fortran
      ncase = 112
      mfmv1 = (/ to_fm('12.1123456789') , to_fm('-34.2123456789') , to_fm('56.3123456789') /)
      div = mfmv1
      mfm3 = 0
      do j = 1, 3
         mfm3 = mfm3 + abs(div(j) - int(mfmv1(j)))
      enddo
      call fm_st2m(' 1.0e-45 ', mfm4)
      if (.not.(mfm3 <= mfm4)) then
          call prterr(kwsave)
      endif

      kw = kwsave
      return
      end subroutine test15

      subroutine test16

!  Test type (im) array equal assignments.

      implicit none
      integer :: j

      kwsave = kw
      call fmsetvar(' kw = 22 ')

      ncase = 113
      mim1 = to_fm('234.56')
      div = mim1
      di5 = 0
      do j = 1, 3
         di5 = di5 + abs(div(j) - 234)
      enddo
      call fm_st2m(' 1.0e-45 ', mfm4)
      if (.not.(to_fm(di5) <= mfm4)) then
          call prterr(kwsave)
      endif

      ncase = 114
      div = (/ 12, -34, 56 /)
      mimv1 = div
      mfm3 = 0
      do j = 1, 3
         mfm3 = mfm3 + abs(mimv1(j) - div(j))
      enddo
      call fm_st2m(' 1.0e-45 ', mfm4)
      if (.not.(mfm3 <= mfm4)) then
          call prterr(kwsave)
      endif

      ncase = 115
      rv = (/ 12.1, -34.2, 56.3 /)
      mimv1 = rv
      mfm3 = 0
      div = rv
      do j = 1, 3
         mfm3 = mfm3 + abs(mimv1(j) - div(j))
      enddo
      enddo
```