

```
! This is a test program for version 1.4 of module FM_QUAD_REAL, 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_QUAD_REAL are tested, and if all tests are completed
! successfully, this line is printed:
```

```
! 512 cases tested. No errors were found.
```

```
MODULE TEST_VARS
```

```
USE FMVALS
```

```
USE FMZM
```

```
USE FM_QUAD_REAL
```

```
!           Declare the derived type variables of type (FM), (IM), and (ZM).
!           These are in the form that would be found in a user program.
```

```
TYPE (FM), SAVE :: M_A, MFM1, MFM2, MFM3, MFM4, MFM5, 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)
```

```
!           These are the variables that are not multiple precision.
```

```
INTEGER, SAVE :: JV(3), JV2(3,3)
```

```
REAL, SAVE :: R3, RSMALL
```

```
REAL (QUAD_FP), SAVE :: QD1, QD2, QD3, QD4, QD5, QDS, QDV(3), QDM(3,3)
```

```
COMPLEX, SAVE :: C3
```

```
COMPLEX (QUAD_FP), SAVE :: ZQ1, ZQ2, ZQ3, ZQ4, ZQV(3), ZQM(3,3)
```

```
INTEGER, SAVE :: J, K, 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.')")
```

QDS = EPSILON(Q_ONE)*100.0

NCASE = 1

QD4 = MFM1

IF (ABS((QD4-581.21_QUAD_FP)/581.21_QUAD_FP) > QDS) CALL PRterr(KW)

NCASE = 2

QD4 = MIM1

IF (ABS((QD4-661.0_QUAD_FP)/661.0_QUAD_FP) > QDS) CALL PRterr(KW)

NCASE = 3

QD4 = MZM1

IF (ABS((QD4-731.51_QUAD_FP)/731.51_QUAD_FP) > QDS) CALL PRterr(KW)

NCASE = 4

ZQ4 = MFM1

IF (ABS((ZQ4-581.21_QUAD_FP)/581.21_QUAD_FP) > QDS) CALL PRterr(KW)

NCASE = 5

ZQ4 = MIM1

IF (ABS((ZQ4-661.0_QUAD_FP)/661.0_QUAD_FP) > QDS) CALL PRterr(KW)

NCASE = 6

ZQ4 = MZM1

IF (ABS((ZQ4-(731.51_QUAD_FP,711.41_QUAD_FP))/(731.51_QUAD_FP,711.41_QUAD_FP)) > QDS) &
CALL PRterr(KW)

NCASE = 7

MFM3 = QD2

CALL FM_ST2M('391.6123456789012345678901',MFM4)

CALL FM_SUB(MFM3,MFM4,MFM6)

CALL FM_EQ(MFM6,MFM4)

CALL FM_DIV(MFM4,MFM3,MFM6)

CALL FM_EQ(MFM6,MFM4)

CALL FM_ABS(MFM4,MFM6)

CALL FM_EQ(MFM6,MFM4)

MFM3 = QDS

IF (FM_COMP(MFM4,'GT',MFM3)) CALL PRterr(KW)

NCASE = 8

MFM3 = ZQ2

CALL FM_ST2M('431.11',MFM4)

CALL FM_SUB(MFM3,MFM4,MFM6)

CALL FM_EQ(MFM6,MFM4)

CALL FM_DIV(MFM4,MFM3,MFM6)

CALL FM_EQ(MFM6,MFM4)

CALL FM_ABS(MFM4,MFM6)

CALL FM_EQ(MFM6,MFM4)

MFM3 = QDS

IF (FM_COMP(MFM4,'GT',MFM3)) CALL PRterr(KW)

NCASE = 9

MFM3 = TO_FM(ZQ2)

CALL FM_ST2M('431.11',MFM4)

CALL FM_SUB(MFM3,MFM4,MFM6)

CALL FM_EQ(MFM6,MFM4)

CALL FM_DIV(MFM4,MFM3,MFM6)

CALL FM_EQ(MFM6,MFM4)

```
CALL FM_ABS(MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
MFM3 = QDS
IF (FM_COMP(MFM4, 'GT',MFM3)) CALL PRTErr(KW)
```

```
NCASE = 10
MIM3 = QD2
CALL IM_ST2M('391',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 = ZQ2
CALL IM_ST2M('431',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 = 12
MIM3 = TO_IM(ZQ2)
CALL IM_ST2M('431',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 = 13
MZM3 = QD2
CALL ZM_ST2M('391.6123456789012345678901',MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
MFM3 = QDS
IF (FM_COMP(MFM4, 'GT',MFM3)) CALL PRTErr(KW)
```

```
NCASE = 14
MZM3 = ZQ2
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
MFM3 = QDS
IF (FM_COMP(MFM4, 'GT',MFM3)) CALL PRTErr(KW)
```

```
NCASE = 15
MZM3 = TO_ZM(ZQ2)
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
```

```
CALL FM_DIV(MFM5,MFM6,MFM4)
MFM3 = QDS
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
```

```
QD1 = 12.345678901234567890123_QUAD_FP
```

```
M_A = QD1
```

```
IF (.NOT.(M_A == QD1)) THEN
```

```
CALL ERRPRT_FM(' == ',M_A, 'M_A',M_A, 'M_A',M_A, 'M_A')
```

```
ENDIF
```

```
NCASE = 17
```

```
QD1 = 12.345678901234567890123_QUAD_FP
```

```
M_A = QD1
```

```
IF (.NOT.(QD1 == M_A)) THEN
```

```
CALL ERRPRT_FM(' == ',M_A, 'M_A',M_A, 'M_A',M_A, 'M_A')
```

```
ENDIF
```

```
NCASE = 18
```

```
QD1 = 123
```

```
M_J = QD1
```

```
IF (.NOT.(M_J == QD1)) THEN
```

```
CALL ERRPRT_IM(' == ',M_J, 'M_J',M_J, 'M_J')
```

```
ENDIF
```

```
NCASE = 19
```

```
QD1 = 123
```

```
M_J = QD1
```

```
IF (.NOT.(QD1 == M_J)) THEN
```

```
CALL ERRPRT_IM(' == ',M_J, 'M_J',M_J, 'M_J')
```

```
ENDIF
```

```
NCASE = 20
```

```
QD1 = 12.345678901234567890123_QUAD_FP
```

```
M_Z = QD1
```

```
IF (.NOT.(M_Z == QD1)) THEN
```

```
CALL ERRPRT_ZM(' == ',M_Z, 'M_Z',M_Z, 'M_Z',M_Z, 'M_Z')
```

```
ENDIF
```

```
NCASE = 21
```

```
QD1 = 12.345678901234567890123_QUAD_FP
```

```
M_Z = QD1
```

```
IF (.NOT.(QD1 == M_Z)) THEN
```

```
CALL ERRPRT_ZM(' == ',M_Z, 'M_Z',M_Z, 'M_Z',M_Z, 'M_Z')
```

```
ENDIF
```

```
NCASE = 22
```

```
ZQ1 = 12.3
```

```
M_A = ZQ1
IF (.NOT.(M_A == ZQ1)) THEN
  CALL ERRPRT_FM(' == ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF
```

```
NCASE = 23
ZQ1 = (12.3 , 45.6)
M_A = ZQ1
IF (M_A == ZQ1) THEN
  CALL ERRPRT_FM(' == ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF
```

```
NCASE = 24
ZQ1 = 12.3
M_A = ZQ1
IF (.NOT.(ZQ1 == M_A)) THEN
  CALL ERRPRT_FM(' == ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF
```

```
NCASE = 25
ZQ1 = (12.3 , 45.6)
M_A = ZQ1
IF (ZQ1 == M_A) THEN
  CALL ERRPRT_FM(' == ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF
```

```
NCASE = 26
ZQ1 = 123
M_J = ZQ1
IF (.NOT.(M_J == ZQ1)) THEN
  CALL ERRPRT_IM(' == ',M_J,'M_J',M_J,'M_J')
ENDIF
```

```
NCASE = 27
ZQ1 = (123.0 , 45.6)
M_J = ZQ1
IF (M_J == ZQ1) THEN
  CALL ERRPRT_IM(' == ',M_J,'M_J',M_J,'M_J')
ENDIF
```

```
NCASE = 28
ZQ1 = 123
M_J = ZQ1
IF (.NOT.(ZQ1 == M_J)) THEN
  CALL ERRPRT_IM(' == ',M_J,'M_J',M_J,'M_J')
ENDIF
```

```
NCASE = 29
ZQ1 = (123.0 , 45.6)
M_J = ZQ1
IF (ZQ1 == M_J) THEN
  CALL ERRPRT_IM(' == ',M_J,'M_J',M_J,'M_J')
ENDIF
```

```
NCASE = 30
ZQ1 = (12.3 , 45.6)
M_Z = ZQ1
IF (.NOT.(M_Z == ZQ1)) THEN
```

```

    CALL ERRPRT_ZM(' == ',M_Z,'M_Z',M_Z,'M_Z',M_Z,'M_Z')
ENDIF

NCASE = 31
ZQ1 = (12.3 , 45.6)
M_Z = ZQ1
IF (.NOT.(ZQ1 == 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 = 32
QD1 = 12.345678901234567890123_QUAD_FP
M_A = 1 + QD1
IF (.NOT.(M_A /= QD1)) THEN
    CALL ERRPRT_FM(' /= ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF

```

```

NCASE = 33
QD1 = 12.345678901234567890123_QUAD_FP
M_A = 1 + QD1
IF (.NOT.(QD1 /= M_A)) THEN
    CALL ERRPRT_FM(' /= ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF

```

```

NCASE = 34
QD1 = 123
M_J = 1 + QD1
IF (.NOT.(M_J /= QD1)) THEN
    CALL ERRPRT_IM(' /= ',M_J,'M_J',M_J,'M_J')
ENDIF

```

```

NCASE = 35
QD1 = 123
M_J = 1 + QD1
IF (.NOT.(QD1 /= M_J)) THEN
    CALL ERRPRT_IM(' /= ',M_J,'M_J',M_J,'M_J')
ENDIF

```

```

NCASE = 36
QD1 = 12.345678901234567890123_QUAD_FP
M_Z = 1 + QD1
IF (.NOT.(M_Z /= QD1)) THEN
    CALL ERRPRT_ZM(' /= ',M_Z,'M_Z',M_Z,'M_Z',M_Z,'M_Z')
ENDIF

```

```

NCASE = 37
QD1 = 12.345678901234567890123_QUAD_FP

```

```
M_Z = ( 12.3 , 34.5 )
IF (.NOT.(M_Z /= QD1)) THEN
    CALL ERRPRT_ZM(' /= ',M_Z,'M_Z',M_Z,'M_Z',M_Z,'M_Z')
ENDIF
```

```
NCASE = 38
QD1 = 12.345678901234567890123_QUAD_FP
M_Z = 1 + QD1
IF (.NOT.(QD1 /= M_Z)) THEN
    CALL ERRPRT_ZM(' /= ',M_Z,'M_Z',M_Z,'M_Z',M_Z,'M_Z')
ENDIF
```

```
NCASE = 39
QD1 = 12.345678901234567890123_QUAD_FP
M_Z = ( 12.3 , 34.5 )
IF (.NOT.(QD1 /= M_Z)) THEN
    CALL ERRPRT_ZM(' /= ',M_Z,'M_Z',M_Z,'M_Z',M_Z,'M_Z')
ENDIF
```

```
NCASE = 40
ZQ1 = 12.3
M_A = 1 + ZQ1
IF (.NOT.(M_A /= ZQ1)) THEN
    CALL ERRPRT_FM(' /= ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF
```

```
NCASE = 41
ZQ1 = (12.3 , 45.6)
M_A = (12.3 , 45.6)
IF (.NOT.(M_A /= ZQ1)) THEN
    CALL ERRPRT_FM(' /= ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF
```

```
NCASE = 42
ZQ1 = 12.3
M_A = 1 + ZQ1
IF (.NOT.(ZQ1 /= M_A)) THEN
    CALL ERRPRT_FM(' /= ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF
```

```
NCASE = 43
ZQ1 = (12.3 , 45.6)
M_A = (12.3 , 45.6)
IF (.NOT.(ZQ1 /= M_A)) THEN
    CALL ERRPRT_FM(' /= ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF
```

```
NCASE = 44
ZQ1 = 123
M_J = 1 + ZQ1
IF (.NOT.(M_J /= ZQ1)) THEN
    CALL ERRPRT_IM(' /= ',M_J,'M_J',M_J,'M_J')
ENDIF
```

```
NCASE = 45
ZQ1 = (123.0 , 45.6)
M_J = (123.0 , 45.6)
IF (.NOT.(M_J /= ZQ1)) THEN
```

```

    CALL ERRPRT_IM(' /= ',M_J,'M_J',M_J,'M_J')
ENDIF

NCASE = 46
ZQ1 = 123
M_J = 1 + ZQ1
IF (.NOT.(ZQ1 /= M_J)) THEN
    CALL ERRPRT_IM(' /= ',M_J,'M_J',M_J,'M_J')
ENDIF

NCASE = 47
ZQ1 = (123.0 , 45.6)
M_J = (123.0 , 45.6)
IF (.NOT.(ZQ1 /= M_J)) THEN
    CALL ERRPRT_IM(' /= ',M_J,'M_J',M_J,'M_J')
ENDIF

NCASE = 48
ZQ1 = (12.3 , 45.6)
M_Z = 1 + ZQ1
IF (.NOT.(M_Z /= ZQ1)) THEN
    CALL ERRPRT_ZM(' /= ',M_Z,'M_Z',M_Z,'M_Z',M_Z,'M_Z')
ENDIF

NCASE = 49
ZQ1 = (12.3 , 45.6)
M_Z = 1 + ZQ1
IF (.NOT.(ZQ1 /= 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 = 50
QD1 = 12.345678901234567890123_QUAD_FP
M_A = QD1 + 1
IF (.NOT.(M_A > QD1)) THEN
    CALL ERRPRT_FM(' > ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF

NCASE = 51
QD1 = 12.345678901234567890123_QUAD_FP
M_A = QD1 - 1
IF (.NOT.(QD1 > M_A)) THEN
    CALL ERRPRT_FM(' > ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF

NCASE = 52
QD1 = 123

```



```

M_J = QD1 + 1
IF (.NOT.(M_J > QD1)) THEN
  CALL ERRPRT_IM(' > ',M_J,'M_J',M_J,'M_J')
ENDIF

NCASE = 53
QD1 = 123
M_J = QD1 - 1
IF (.NOT.(QD1 > M_J)) THEN
  CALL ERRPRT_IM(' > ',M_J,'M_J',M_J,'M_J')
ENDIF

RETURN
END SUBROUTINE TEST4

SUBROUTINE TEST5

```

! Test the derived type >= interface.

```

IMPLICIT NONE

WRITE (KW, "(/' Testing the derived type >= interface.')" )

NCASE = 54
QD1 = 12.3456789012345678901234567890123_QUAD_FP
M_A = QD1 + 1
IF (.NOT.(M_A >= QD1)) THEN
  CALL ERRPRT_FM(' >= ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF

NCASE = 55
QD1 = 12.345678901234567890123_QUAD_FP
M_A = QD1 - 1
IF (.NOT.(QD1 >= M_A)) THEN
  CALL ERRPRT_FM(' >= ',M_A,'M_A',M_A,'M_A',M_A,'M_A')
ENDIF

NCASE = 56
QD1 = 123
M_J = QD1 + 1
IF (.NOT.(M_J >= QD1)) THEN
  CALL ERRPRT_IM(' >= ',M_J,'M_J',M_J,'M_J')
ENDIF

NCASE = 57
QD1 = 123
M_J = QD1 - 1
IF (.NOT.(QD1 >= 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 = 58
```

```
QD1 = 12.345678901234567890123_QUAD_FP
```

```
M_A = QD1 - 2
```

```
IF (.NOT.(M_A < QD1)) THEN
```

```
    CALL ERRPRT_FM(' < ', M_A, 'M_A', M_A, 'M_A', M_A, 'M_A')
```

```
ENDIF
```

```
NCASE = 59
```

```
QD1 = 12.345678901234567890123_QUAD_FP
```

```
M_A = QD1 + 2
```

```
IF (.NOT.(QD1 < M_A)) THEN
```

```
    CALL ERRPRT_FM(' < ', M_A, 'M_A', M_A, 'M_A', M_A, 'M_A')
```

```
ENDIF
```

```
NCASE = 60
```

```
QD1 = 123
```

```
M_J = QD1 - 2
```

```
IF (.NOT.(M_J < QD1)) THEN
```

```
    CALL ERRPRT_IM(' < ', M_J, 'M_J', M_J, 'M_J')
```

```
ENDIF
```

```
NCASE = 61
```

```
QD1 = 123
```

```
M_J = QD1 + 2
```

```
IF (.NOT.(QD1 < 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 = 62
```

```
QD1 = 12.345678901234567890123_QUAD_FP
```

```
M_A = QD1 - 2
```

```
IF (.NOT.(M_A <= QD1)) THEN
```

```
    CALL ERRPRT_FM(' <= ', M_A, 'M_A', M_A, 'M_A', M_A, 'M_A')
```

```
ENDIF
```

```
NCASE = 63
```

```
QD1 = 12.345678901234567890123_QUAD_FP
```

```
M_A = QD1 + 2
```

```
IF (.NOT.(QD1 <= M_A)) THEN
```

```
    CALL ERRPRT_FM(' <= ', M_A, 'M_A', M_A, 'M_A', M_A, 'M_A')
```

```
ENDIF
```

```
NCASE = 64
```

```

QD1 = 123
M_J = QD1 - 2
IF (.NOT.(M_J <= QD1)) THEN
    CALL ERRPRT_IM(' <= ',M_J,'M_J',M_J,'M_J')
ENDIF

NCASE = 65
QD1 = 123
M_J = QD1 + 2
IF (.NOT.(QD1 <= M_J)) THEN
    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.')" )

```

```

QDS = EPSILON(Q_ONE)*100.0

```

```

NCASE = 66
MFM3 = QD2 + MFM1
CALL FM_ST2M('391.6123456789012345678901',MFM4)
CALL FM_ADD(MFM4,MFM1,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_SUB(MFM3,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_DIV(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_ABS(MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)

```

```

NCASE = 67
CALL FM_ST2M('391.6123456789012345678901',MFM4)
CALL FM_ST2M('661',MFM3)
CALL FM_ADD(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
MFM3 = QD2 + MFM1
CALL FM_SUB(MFM3,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_DIV(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_ABS(MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)

```

```

NCASE = 68
MFM3 = QD2 + MFM1
CALL ZM_ST2M('391.6123456789012345678901',MFM4)
CALL ZM_ADD(MFM4,MFM1,MFM5)
CALL ZM_EQ(MFM5,MFM4)

```

```
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 69

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_ST2M('581.21',MZM3)
CALL ZM_ADD(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = ZQ2 + MFM1
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 70

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_ST2M('661',MZM3)
CALL ZM_ADD(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = ZQ2 + MIM1
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 71

```
MZM3 = ZQ2 + MZM1
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_ADD(MZM4,MZM1,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 72

```
MFM3 = MFM1 + QD2
CALL FM_ST2M('391.6123456789012345678901',MFM4)
CALL FM_ADD(MFM1,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_SUB(MFM3,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_DIV(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_ABS(MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 73

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM3)
CALL ZM_ST2M('581.21',MZM4)
CALL ZM_ADD(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = MFM1 + ZQ2
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTER(KW)
```

NCASE = 74

```
CALL FM_ST2M('391.6123456789012345678901',MFM3)
CALL FM_ST2M('661',MFM4)
CALL FM_ADD(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
MFM3 = MIM1 + QD2
CALL FM_SUB(MFM3,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_DIV(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_ABS(MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTER(KW)
```

NCASE = 75

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM3)
CALL ZM_ST2M('661',MZM4)
CALL ZM_ADD(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = MIM1 + ZQ2
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTER(KW)
```

NCASE = 76

```
MZM3 = MZM1 + QD2
CALL ZM_ST2M('391.6123456789012345678901',MZM4)
CALL ZM_ADD(MZM1,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTER(KW)
```

NCASE = 77

```
MZM3 = MZM1 + ZQ2
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_ADD(MZM1,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
```

```
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

```
END SUBROUTINE TEST8
```

```
SUBROUTINE TEST9
```

! Test the '-' arithmetic operator.

```
IMPLICIT NONE
```

```
WRITE (KW, "(/' Testing the derived type - interface.')")
```

```
QDS = EPSILON(Q_ONE)*100.0
```

```
NCASE = 78
```

```
MFM3 = QD2 - MFM1
```

```
CALL FM_ST2M('391.6123456789012345678901',MFM4)
```

```
CALL FM_SUB(MFM4,MFM1,MFM6)
```

```
CALL FM_EQ(MFM6,MFM4)
```

```
CALL FM_SUB(MFM3,MFM4,MFM6)
```

```
CALL FM_EQ(MFM6,MFM4)
```

```
CALL FM_DIV(MFM4,MFM3,MFM6)
```

```
CALL FM_EQ(MFM6,MFM4)
```

```
CALL FM_ABS(MFM4,MFM6)
```

```
CALL FM_EQ(MFM6,MFM4)
```

```
IF (MFM4 > QDS) CALL PRterr(KW)
```

```
NCASE = 79
```

```
CALL FM_ST2M('391.6123456789012345678901',MFM4)
```

```
CALL FM_ST2M('661',MFM3)
```

```
CALL FM_SUB(MFM4,MFM3,MFM6)
```

```
CALL FM_EQ(MFM6,MFM4)
```

```
MFM3 = QD2 - MIM1
```

```
CALL FM_SUB(MFM3,MFM4,MFM6)
```

```
CALL FM_EQ(MFM6,MFM4)
```

```
CALL FM_DIV(MFM4,MFM3,MFM6)
```

```
CALL FM_EQ(MFM6,MFM4)
```

```
CALL FM_ABS(MFM4,MFM6)
```

```
CALL FM_EQ(MFM6,MFM4)
```

```
IF (MFM4 > QDS) CALL PRterr(KW)
```

```
NCASE = 80
```

```
MZM3 = QD2 - MZM1
```

```
CALL ZM_ST2M('391.6123456789012345678901',MZM4)
```

```
CALL ZM_SUB(MZM4,MZM1,MZM5)
```

```
CALL ZM_EQ(MZM5,MZM4)
```

```
CALL ZM_SUB(MZM3,MZM4,MZM5)
```

```
CALL ZM_EQ(MZM5,MZM4)
```

```
CALL ZM_ABS(MZM4,MFM5)
```

```
CALL ZM_ABS(MZM3,MFM6)
```

```
CALL FM_DIV(MFM5,MFM6,MFM4)
```

```
IF (MFM4 > QDS) CALL PRterr(KW)
```

```
NCASE = 81
```

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_ST2M('581.21',MZM3)
CALL ZM_SUB(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = ZQ2 - MFM1
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 82

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_ST2M('661',MZM3)
CALL ZM_SUB(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = ZQ2 - MIM1
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 83

```
MZM3 = ZQ2 - MZM1
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_SUB(MZM4,MZM1,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 84

```
MFM3 = MFM1 - QD2
CALL FM_ST2M('391.6123456789012345678901',MFM4)
CALL FM_SUB(MFM1,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_SUB(MFM3,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_DIV(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_ABS(MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 85

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM3)
CALL ZM_ST2M('581.21',MZM4)
CALL ZM_SUB(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = MFM1 - ZQ2
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
```

```
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 86

```
CALL FM_ST2M('391.6123456789012345678901',MFM3)
CALL FM_ST2M('661',MFM4)
CALL FM_SUB(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
MFM3 = MIM1 - QD2
CALL FM_SUB(MFM3,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_DIV(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_ABS(MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 87

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM3)
CALL ZM_ST2M('661',MZM4)
CALL ZM_SUB(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = MIM1 - ZQ2
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 88

```
MZM3 = MZM1 - QD2
CALL ZM_ST2M('391.6123456789012345678901',MZM4)
CALL ZM_SUB(MZM1,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRterr(KW)
```

NCASE = 89

```
MZM3 = MZM1 - ZQ2
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_SUB(MZM1,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) 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.')")
```

```
QDS = EPSILON(Q_ONE)*100.0
```

```
NCASE = 90
```

```
MFM3 = QD2 * MFM1
```

```
CALL FM_ST2M('391.6123456789012345678901', MFM4)
```

```
CALL FM_MPY(MFM4, MFM1, MFM6)
```

```
CALL FM_EQ(MFM6, MFM4)
```

```
CALL FM_SUB(MFM3, MFM4, MFM6)
```

```
CALL FM_EQ(MFM6, MFM4)
```

```
CALL FM_DIV(MFM4, MFM3, MFM6)
```

```
CALL FM_EQ(MFM6, MFM4)
```

```
CALL FM_ABS(MFM4, MFM6)
```

```
CALL FM_EQ(MFM6, MFM4)
```

```
IF (MFM4 > QDS) CALL PRterr(KW)
```

```
NCASE = 91
```

```
CALL FM_ST2M('391.6123456789012345678901', MFM4)
```

```
CALL FM_ST2M('661', MFM3)
```

```
CALL FM_MPY(MFM4, MFM3, MFM6)
```

```
CALL FM_EQ(MFM6, MFM4)
```

```
MFM3 = QD2 * MFM1
```

```
CALL FM_SUB(MFM3, MFM4, MFM6)
```

```
CALL FM_EQ(MFM6, MFM4)
```

```
CALL FM_DIV(MFM4, MFM3, MFM6)
```

```
CALL FM_EQ(MFM6, MFM4)
```

```
CALL FM_ABS(MFM4, MFM6)
```

```
CALL FM_EQ(MFM6, MFM4)
```

```
IF (MFM4 > QDS) CALL PRterr(KW)
```

```
NCASE = 92
```

```
MZM3 = QD2 * MZM1
```

```
CALL ZM_ST2M('391.6123456789012345678901', MZM4)
```

```
CALL ZM_MPY(MZM4, MZM1, MZM5)
```

```
CALL ZM_EQ(MZM5, MZM4)
```

```
CALL ZM_SUB(MZM3, MZM4, MZM5)
```

```
CALL ZM_EQ(MZM5, MZM4)
```

```
CALL ZM_ABS(MZM4, MFM5)
```

```
CALL ZM_ABS(MZM3, MFM6)
```

```
CALL FM_DIV(MFM5, MFM6, MFM4)
```

```
IF (MFM4 > QDS) CALL PRterr(KW)
```

```
NCASE = 93
```

```
CALL ZM_ST2M('431.11 + 441.21 i', MZM4)
```

```
CALL ZM_ST2M('581.21',MZM3)
CALL ZM_MPY(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = ZQ2 * MFM1
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTER(KW)
```

NCASE = 94

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_ST2M('661',MZM3)
CALL ZM_MPY(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = ZQ2 * MIM1
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTER(KW)
```

NCASE = 95

```
MZM3 = ZQ2 * MZM1
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_MPY(MZM4,MZM1,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTER(KW)
```

NCASE = 96

```
MFM3 = MFM1 * QD2
CALL FM_ST2M('391.6123456789012345678901',MFM4)
CALL FM_MPY(MFM1,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_SUB(MFM3,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_DIV(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_ABS(MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTER(KW)
```

NCASE = 97

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM3)
CALL ZM_ST2M('581.21',MZM4)
CALL ZM_MPY(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = MFM1 * ZQ2
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
```

```
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTERR(KW)
```

NCASE = 98

```
CALL FM_ST2M('391.6123456789012345678901',MFM3)
CALL FM_ST2M('661',MFM4)
CALL FM_MPY(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
MFM3 = MIM1 * QD2
CALL FM_SUB(MFM3,MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_DIV(MFM4,MFM3,MFM6)
CALL FM_EQ(MFM6,MFM4)
CALL FM_ABS(MFM4,MFM6)
CALL FM_EQ(MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTERR(KW)
```

NCASE = 99

```
CALL ZM_ST2M('431.11 + 441.21 i',MZM3)
CALL ZM_ST2M('661',MZM4)
CALL ZM_MPY(MZM4,MZM3,MZM5)
CALL ZM_EQ(MZM5,MZM4)
MZM3 = MIM1 * ZQ2
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTERR(KW)
```

NCASE = 100

```
MZM3 = MZM1 * QD2
CALL ZM_ST2M('391.6123456789012345678901',MZM4)
CALL ZM_MPY(MZM1,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTERR(KW)
```

NCASE = 101

```
MZM3 = MZM1 * ZQ2
CALL ZM_ST2M('431.11 + 441.21 i',MZM4)
CALL ZM_MPY(MZM1,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_SUB(MZM3,MZM4,MZM5)
CALL ZM_EQ(MZM5,MZM4)
CALL ZM_ABS(MZM4,MFM5)
CALL ZM_ABS(MZM3,MFM6)
CALL FM_DIV(MFM5,MFM6,MFM4)
IF (MFM4 > QDS) CALL PRTERR(KW)
```

END SUBROUTINE TEST10

SUBROUTINE TEST11

! Test the '/' arithmetic operator.

IMPLICIT NONE

WRITE (KW, "(/' Testing the derived type / interface.')")

QDS = EPSILON(Q_ONE)*100.0

NCASE = 102

MFM3 = QD2 / MFM1

CALL FM_ST2M('391.6123456789012345678901', MFM4)

CALL FM_DIV(MFM4, MFM1, MFM6)

CALL FM_EQ(MFM6, MFM4)

CALL FM_SUB(MFM3, MFM4, MFM6)

CALL FM_EQ(MFM6, MFM4)

CALL FM_DIV(MFM4, MFM3, MFM6)

CALL FM_EQ(MFM6, MFM4)

CALL FM_ABS(MFM4, MFM6)

CALL FM_EQ(MFM6, MFM4)

IF (MFM4 > QDS) CALL PRterr(KW)

NCASE = 103

CALL FM_ST2M('391.6123456789012345678901', MFM4)

CALL FM_ST2M('661', MFM3)

CALL FM_DIV(MFM4, MFM3, MFM6)

CALL FM_EQ(MFM6, MFM4)

MFM3 = QD2 / MFM1

CALL FM_SUB(MFM3, MFM4, MFM6)

CALL FM_EQ(MFM6, MFM4)

CALL FM_DIV(MFM4, MFM3, MFM6)

CALL FM_EQ(MFM6, MFM4)

CALL FM_ABS(MFM4, MFM6)

CALL FM_EQ(MFM6, MFM4)

IF (MFM4 > QDS) CALL PRterr(KW)

NCASE = 104

MZM3 = QD2 / MZM1

CALL ZM_ST2M('391.6123456789012345678901', MZM4)

CALL ZM_DIV(MZM4, MZM1, MZM5)

CALL ZM_EQ(MZM5, MZM4)

CALL ZM_SUB(MZM3, MZM4, MZM5)

CALL ZM_EQ(MZM5, MZM4)

CALL ZM_ABS(MZM4, MFM5)

CALL ZM_ABS(MZM3, MFM6)

CALL FM_DIV(MFM5, MFM6, MFM4)

IF (MFM4 > QDS) CALL PRterr(KW)

NCASE = 105

CALL ZM_ST2M('431.11 + 441.21 i', MZM4)

CALL ZM_ST2M('581.21', MZM3)

CALL ZM_DIV(MZM4, MZM3, MZM5)

CALL ZM_EQ(MZM5, MZM4)

MZM3 = ZQ2 / MFM1

CALL ZM_SUB(MZM3, MZM4, MZM5)

CALL ZM_EQ(MZM5, MZM4)

CALL ZM_ABS(MZM4, MFM5)

CALL ZM_ABS(MZM3, MFM6)