

```

(do-test "expt simple function"
(equal (CL:EXPT -1.0 2) 1.0))

(do-test "expt complex number"
(equal (CL:EXPT 0.0 (complex 1.0 -5.0)) 0.0))

(do-test "expt zero"
(equal (CL:EXPT 0.0 0) 1.0))

(do-test "expt fraction raised to the zero"
(eq (CL:EXPT (/ 1 2) 0) 1))

(do-test "expt negative raised to a fraction"
(equal (CL:EXPT -8 (/ 1 4)) #C(1.1892071 1.1892071)))

(do-test "expt very large number"
(equal (CL:EXPT -1.539016e+9 2) 2.3685701e+18))

(do-test "sqrt with complex number"
(equal (sqrt #C(0.0 0.0)) 0.0))

(do-test "asin with complex number"
(equal (asin #C(1.0 0.0)) #C(1.5707964 0.0)))

(do-test "phase with complex number"
(equal (phase #C(1.0 0.0)) 0.0))

(do-test "acosh with complex number"
(equal (acosh #C(-2.9732 -3.328)) #C(2.1905336 -2.2875323) ))

(do-test "rational"
(il:leq (rational -1e20) 0))

(do-test "decode-float and scale-float are inverses"
(and
(setq x 3.8246e-41)
(multiple-value-setq (a b c) (decode-float x))
(equal x (scale-float a b))))

(do-test "Floor and bignums"
(multiple-value-bind (f r) (floor -2165/60893 31072)
(= -2165/60893 (+ r (* f 31072))))
)

(do-test "type expander for Complex"
(and (not (typep #C(5 6) '(complex float)))
(typep #C(5 6) '(complex integer)))
)

```