assert(r) produces the error message “assertion is false” and aborts with error if \( r == 0 \).

asserteq(A, B) is logically equivalent to assert(A==B). If the assertion is false, however, information is presented on the number of mismatches.

### Syntax

```plaintext
void assert(real scalar r)
void asserteq(transmorphic matrix A, transmorphic matrix B)
```

### Remarks and examples

In the midst of complicated code, you know that a certain calculation must produce a result greater than 0, but you worry that perhaps you have an error in your code:

```plaintext
... assert(n>0) ...
```

In another spot, you have produced matrix A and know every element of A should be positive or zero:

```plaintext
... assert(all(A:>=0)) ...
```

Once you are convinced that your function works, these verifications should be removed. In a third part of your code, however, the problem is different if the number of rows \( r \) exceed the number of columns \( c \). In all the cases you need to use it, however, \( r \) will be less than \( c \), so you are not much interested in programming the alternative solution:

```plaintext
... assert(rows(PROBLEM) < cols(PROBLEM)) ...
```

Leave that one in.
Conformability

assert(r):
    \(r\): \(1 \times 1\)
    result: void

asserteq(A, B):
    A: \(r_1 \times c_1\)
    B: \(r_2 \times c_2\)
    result: void

Diagnostics

assert(r) aborts with error if \(r == 0\).

asserteq(A, B) aborts with error if \(A \neq B\).

Also see

[M-4] Programming — Programming functions