



Svigruppo

Monga

Divisione del
lavoro

Assezioni

Lezione XIII: Documentazione dei componenti

assert (3)



Svigruppo

Monga

Divisione del
lavoro

Asserzioni

NAME

`assert` - abort the program if assertion is false

SYNOPSIS

```
#include <assert.h>
```

```
void assert(scalar expression);
```

DESCRIPTION

If the macro `NDEBUG` was defined at the moment `<assert.h>` was last included, the macro `assert()` generates no code, and hence does nothing at all. Otherwise, the macro `assert()` prints an error message to standard error and terminates the program by calling `abort(3)` if expression is false (i.e., compares equal to zero).

CONFORMING TO

POSIX.1-2001, C89, C99. In C89, expression is required to be of type `int`.

BUGS

`assert()` is implemented as a macro; if the expression tested has side-effects, program behavior will be different depending on whether `NDEBUG` is defined. This may create Heisenbugs which go away when debugging is turned on.



Ormai presente in quasi tutti i linguaggi nativo o nelle librerie standard:

Java `assert`

Python `assert`

PHP `assert`

Javascript `console.assert` (non in Explorer...)

...



```
int square_root(int x);  
/*@  
  assume x >= 0;  
  return y where y >= 0;  
  return y where y*y <= x  
    &&& x < (y+1)*(y+1);  
@*/
```

```
void swap(int* x, int* y);  
/*@  
  assume x &&& y &&& x != y;  
  promise *x == in *y;  
  promise *y == in *x;  
@*/  
void swap(int* x, int* y) {  
  *x = *x + *y;  
  *y = *x - *y;  
  /*@ assert *y == in *x; @*/  
  *x = *x - *y;  
}
```

