```
double nearbyint(double x);
float nearbyintf(float x);
long double nearbyintl(long double x);

double rint(double x);
float rintf(float x);
long double rintl(long double x);

long int lrint(double x);
long int lrintf(float x);
long int lrintl(long double x);
long long int llrint(double x);
long long int llrintf(float x);
long long int llrintl(long double x);

double round(double x);
float roundf(float x);
long double roundl(long double x);

long int lround(double x);
long int lroundf(float x);
long int lroundl(long double x);
long long int llround(double x);
long long int llroundf(float x);
long long int llroundl(long double x);

double trunc(double x);
float truncf(float x);
long double truncl(long double x);
```

Besides additional versions of `ceil` and `floor`, C99 has a number of new functions that convert a floating-point value to the nearest integer. Be careful when using these functions: although all of them return an integer, some functions return it in floating-point format (as a `float`, `double`, or `long double` value) and some return it in integer format (as a `long int` or `long long int` value).

*nearbyint* *rint*

The `nearbyint` function rounds its argument to an integer, returning it as a floating-point number. `nearbyint` uses the current rounding direction and does not raise the *inexact* floating-point exception. `rint` is the same as `nearbyint`, except that it may raise the *inexact* floating-point exception if the result has a different value than the argument.

*lrint* *llrint*

The `lrint` function rounds its argument to the nearest integer, according to the current rounding direction. `lrint` returns a `long int` value. `llrint` is the same as `lrint`, except that it returns a `long long int` value.

*round*

The `round` function rounds its argument to the nearest integer value, returning it as a floating-point number. `round` always rounds away from zero (so 3.5 is rounded to 4.0, for example).