tion. May raise the *inexact* floating-point exception if the result has a different value than x. *23.4*

---

round *Round to Nearest Integral Value (C99)* `<math.h>`

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

Returns x rounded to the nearest integer (in floating-point format). Halfway cases are rounded away from zero. *23.4*

---

scalbln *Scale Floating-Point Number Using Long Integer (C99)* `<math.h>`

```
           double scalbln(double x, long int n);
scalblnf   float scalblnf(float x, long int n);
scalblnl   long double scalblnl(long double x, long int n);
```

Returns $x \times$ FLT_RADIX^n, computed in an efficient way. A range error may occur. *23.4*

---

scalbn *Scale Floating-Point Number Using Integer (C99)* `<math.h>`

```
          double scalbn(double x, int n);
scalbnf   float scalbnf(float x, int n);
scalbnl   long double scalbnl(long double x, int n);
```

Returns $x \times$ FLT_RADIX^n, computed in an efficient way. A range error may occur. *23.4*

---

scanf *Formatted Read* `<stdio.h>`

```
int scanf(const char * restrict format, ...);
```

Reads input items from the `stdin` stream. The string pointed to by `format` specifies the format of the items to be read. The arguments that follow `format` point to objects in which the items are to be stored.

Returns Number of input items successfully read and stored. Returns `EOF` if an input failure occurs before any items can be read. *3.2, 22.3*

---

setbuf *Set Buffer* `<stdio.h>`

```
void setbuf(FILE * restrict stream,
            char * restrict buf);
```

If `buf` isn't a null pointer, a call of `setbuf` is equivalent to:

```
(void) setvbuf(stream, buf, _IOFBF, BUFSIZ);
```

Otherwise, it's equivalent to:

```
(void) setvbuf(stream, NULL, _IONBF, 0);
```

22.2

---

setjmp *Prepare for Nonlocal Jump* `<setjmp.h>`

```
int setjmp(jmp_buf env);
```

macro

Stores the current environment in `env` for use in a later call of `longjmp`.

Returns Zero when called directly. Returns a nonzero value when returning from a call of `longjmp`. *24.4*