**memset** *Initialize Memory Block* <string.h>

```
void *memset(void *s, int c, size_t n);
```

Stores c in each of the first n characters of the object pointed to by s.

*Returns* s (a pointer to the object). *23.6*

---

**mktime** *Convert Broken-Down Local Time to Calendar Time* <time.h>

```
time_t mktime(struct tm *timeptr);
```

Converts a broken-down local time (stored in the structure pointed to by timeptr) into a calendar time. The members of the structure aren't required to be within their legal ranges; also, the values of tm_wday (day of the week) and tm_yday (day of the year) are ignored. mktime stores values in tm_wday and tm_yday after adjusting the other members to bring them into their proper ranges.

*Returns* A calendar time corresponding to the structure pointed to by timeptr. Returns (time_t)(-1) if the calendar time can't be represented. *26.3*

---

**modf** *Split into Integer and Fractional Parts* <math.h>

```
double modf(double value, double *iptr);
float modff(float value, float *iptr);
long double modfl(long double value, long double *iptr);
```

Splits value into integer and fractional parts; stores the integer part in the object pointed to by iptr.

*Returns* Fractional part of value. *23.3*

---

**nan** *Create NaN (C99)* <math.h>

```
double nan(const char *tagp);
float nanf(const char *tagp);
long double nanl(const char *tagp);
```

*Returns* A "quiet" NaN whose binary pattern is determined by the string pointed to by tagp. Returns zero if quiet NaNs aren't supported. *23.4*

---

**nearbyint** *Round to Integral Value Using Current Direction (C99)* <math.h>

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

*Returns* x rounded to an integer (in floating-point format) using the current rounding direction. Doesn't raise the *inexact* floating-point exception. *23.4*

---

**nextafter** *Next Number After (C99)* <math.h>

```
double nextafter(double x, double y);
float nextafterf(float x, float y);
long double nextafterl(long double x, long double y);
```