```
t.tm_sec = 0;
t.tm_min = 0;
t.tm_hour = 0;
t.tm_isdst = -1;
```

Next, we'll add 16 to the tm_mday member:

```
t.tm_mday += 16;
```

That leaves 43 in tm_mday, which is out of range for that member. Calling mktime will bring the members of the structure back into their proper ranges:

```
mktime(&t);
```

We'll discard mktime's return value, since we're interested only in the function's effect on t. The relevant members of t now have the following values:

| *Member* | *Value* | *Meaning* |
|---|---|---|
| tm_mday | 12 | 12 |
| tm_mon | 7 | August |
| tm_year | 112 | 2012 |
| tm_wday | 0 | Sunday |
| tm_yday | 224 | 225th day of the year |

## Time Conversion Functions

```
char *asctime(const struct tm *timeptr);
char *ctime(const time_t *timer);
struct tm *gmtime(const time_t *timer);
struct tm *localtime(const time_t *timer);
size_t strftime(char * restrict s, size_t maxsize,
                const char * restrict format,
                const struct tm * restrict timeptr);
```

The time conversion functions make it possible to convert calendar times to broken-down times. They can also convert times (calendar or broken-down) to string form. The following figure shows how these functions are related: