

```

        case 3: case 23:
            printf("rd"); break;
        default: printf("th"); break;
    }
    printf(" day of ");

    switch (month) {
        case 1: printf("January"); break;
        case 2: printf("February"); break;
        case 3: printf("March"); break;
        case 4: printf("April"); break;
        case 5: printf("May"); break;
        case 6: printf("June"); break;
        case 7: printf("July"); break;
        case 8: printf("August"); break;
        case 9: printf("September"); break;
        case 10: printf("October"); break;
        case 11: printf("November"); break;
        case 12: printf("December"); break;
    }

    printf(", 20%.2d.\n", year);
    return 0;
}

```

Note the use of `%.2d` to display the last two digits of the year. If we had used `%d` instead, single-digit years would be displayed incorrectly (2005 would be printed as 205).

Q & A

Q: My compiler doesn't give a warning when I use `=` instead of `==`. Is there some way to force the compiler to notice the problem? [p. 77]

A: Here's a trick that some programmers use: instead of writing

```
if (i == 0) ...
```

they habitually write

```
if (0 == i) ...
```

Now suppose that the `==` operator is accidentally written as `=`:

```
if (0 = i) ...
```

The compiler will produce an error message, since it's not possible to assign a value to 0. I don't use this trick, because I think it makes programs look unnatural. Also, it can be used only when one of the operands in the test condition isn't an lvalue.

Fortunately, many compilers are capable of checking for suspect uses of the `=` operator in `if` conditions. The GCC compiler, for example, will perform this