Since deliberately falling through from one case into the next is rare, it's a good idea to point out any deliberate omission of `break`:

```
switch (grade) {
  case 4: case 3: case 2: case 1:
          num_passing++;
          /* FALL THROUGH */
  case 0:  total_grades++;
          break;
}
```

Without the comment, someone might later fix the "error" by adding an unwanted `break` statement.

Although the last case in a `switch` statement never needs a `break` statement, it's common practice to put one there anyway to guard against a "missing `break`" problem if cases should later be added.

PROGRAM

Printing a Date in Legal Form

Contracts and other legal documents are often dated in the following way:

Dated this _____ *day of* _____, *20*__.

Let's write a program that displays dates in this form. We'll have the user enter the date in month/day/year form, then we'll display the date in "legal" form:

```
Enter date (mm/dd/yy): 7/19/14
Dated this 19th day of July, 2014.
```

We can get `printf` to do most of the formatting. However, we're left with two problems: how to add "th" (or "st" or "nd" or "rd") to the day, and how to print the month as a word instead of a number. Fortunately, the `switch` statement is ideal for both situations; we'll have one `switch` print the day suffix and another print the month name.

date.c

```
/* Prints a date in legal form */

#include <stdio.h>

int main(void)
{
int month, day, year;

printf("Enter date (mm/dd/yy): ");
scanf("%d /%d /%d", &month, &day, &year);

printf("Dated this %d", day);
switch (day) {
  case 1: case 21: case 31:
    printf("st"); break;
  case 2: case 22:
    printf("nd"); break;
```