Enumerated Input

Enumerated input is more difficult than output, because:

- You have to decide how you want to **encode** the external version.
- You cannot use the input operator (without overloading).
- Because the external representation is text, you can't use a `switch`, but have to use sequential `if`..`else if`..`else` statements instead.

```
Month m1;
cin >> m1; // does not compile
```

Here's a version of a `read_month` function which reads from input, using the first three letters of the input to convert it to a `Month`:

```
istream& read_month(Month& m, istream& in = cin)
{
    string encoded;
    in >> encoded;
    string subs = encoded.substr(0, 3);   // first three characters
    for (char& c : subs) c = to_lower(c); // lowercase

    if (subs == "jan")      m = Month::jan;
    else if (subs == "feb") m = Month::feb;
    else if (subs == "mar") m = Month::mar;
    . . .
    else in.setstate(ios::failbit);   // set failbit
    return in;
}
```

You'd use the function like this:

```
cout << "Enter a month: ";
Month m;
if (read_month(m)) {
    cout << "The month is " << to_string(m) << endl;
}
else {
    cout << "You entered an invalid value." << endl;
}
```