cin >> ch;
while (ch != '#')
{
    cout << ch;
    count++;
    cin >> ch;
}

You can use an int ch, replace cin with cin.get(), cout with cout.put(), and '#' with EOF:

int ch;      /// for compatibility with EOF value
ch = cin.get();
while (ch != EOF)
{
    cout.put(ch);   // cout.put(char(ch)) for some implementations
    count++;
    ch = cin.get();
}

If ch is a character, the loop displays it. If ch is EOF, the loop terminates.

Tip

You should realize that EOF does not represent a character in the input. Instead, it's a signal that there are no more characters.

There's a subtle but important point about using cin.get() beyond the changes made so far. Because EOF represents a value outside the valid character codes, it's possible that it might not be compatible with the char type. For example, on some systems type char is unsigned, so a char variable could never have the usual EOF value of −1. For this reason, if you use cin.get() (no argument) and test for EOF, you must assign the return value to type int instead of type char. Also, if you make ch type int instead of type char, you might have to do a type cast to char when displaying ch.

Listing 5.18 incorporates the cin.get() approach into a new version of Listing 5.15. It also condenses the code by combining character input with the while loop test.