it is a newline character, then the whole line was read. If it isn't a newline character, there are more characters left on the line. This program then uses a loop to reject the rest of the line, but you can modify the code to use the rest of the line for the next input cycle. The program also uses the fact that attempting to read an empty line with `get(char *, int)` causes `cin` to test as `false`.

**Listing 9.8 static.cpp**

```cpp
// static.cpp -- using a static local variable
#include <iostream>
using namespace std;
// constants
const int ArSize = 10;

// function prototype
void strcount(const char * str);

int main()
{
    char input[ArSize];
    char next;

    cout << "Enter a line:\n";
    cin.get(input, ArSize);
    while (cin)
    {
        cin.get(next);
        while (next != '\n') // string didn't fit!
        {
            cin.get(next);
            strcount(input);
            cout << "Enter next line (empty line to quit):\n";
            cin.get(input, ArSize);
        }
    
    cout << "Bye\n";
    return 0;
}
```