

```
#include <iostream>
#include <fstream.h>
void main()
{
    // Datei f. Eingabe & zum "Anhaengen" oeffnen
    fstream inout("test.txt", ios::in | ios::app);
    int cnt=0;
    char ch;
    // aktuelle Position = File-Anfang
    inout.seekg(0, ios::beg);
    while (inout.get(ch)) {
        cout.put(ch);
        cnt++;
        // Wenn Zeilenende gelesen: aktuelle Position
        // am Ende der Datei anhaengen
        if (ch=='\n') {
            streampos mark = inout.tellg();
            inout << cnt << ' ';
            // Wieder zur "alten" Position
            inout.seekg(mark);
        }
    }
    // eof trat auf, daher erfolgen keine Ein-/Ausgaben
    // mehr Status muss "geloescht" werden -> clear
    inout.clear();
    inout << cnt << endl;
    cout << "[ " << cnt << " ]" << endl;
}
```

/*

Einige Kommentare zum Programm:

- inout wird als Ein-/Ausgabedatei geöffnet. Alle Schreiboperationen erfolgen automatisch am Dateiende (Modus: app).
- Um die Datei von Anfang an zu verarbeiten, wird die Position des fiktiven Lesekopfs auf den Dateianfang gesetzt (seekg(0, ios::beg).
- Wenn ein Zeilenumbruch gelesen wurde, wird die aktuelle Position sowie ein Leerzeichen geschrieben. Die aktuelle Position in der Datei wird über die Methode tellg ermittelt.
- Um die Datei nach den Schreiboperationen (die ja am Dateiende erfolgen) weiter zu verarbeiten, wird die Position anschließend wieder hergestellt.
- Die while-Schleife wird verlassen, wenn kein Zeichen mehr gelesen werden kann. Da auftretende Fehler und das Überlesen des letzten Zeichens einer Datei dazu führen, daß keine weiteren Lese- und Schreiboperationen mehr stattfinden, werden die Status-Flags durch clear wieder gelöscht.

*/