



COMMA-SEPARATED VALUES

A Regular Expression Example



CSV EXAMPLE INPUT

- W12345678,Cranston Snort,cs@mail.weber.edu
- W12345678,"Snort, Cranston",cs@mail.weber.edu
- W12345678,Cranston Snort,cs@mail.weber.edu,"a,b"
- W12345678,Cranston""Snort,cs@mail.weber.edu
- W12345678,Cranston Snort,cs@mail.weber.edu,,Room 222

- W12345678,"Snort, Cranston.,cs@mail.weber.edu
- "W12345678","Snort, Cranston,cs@mail.weber.edu



THE CSV PROGRAM DRIVER

```
int main()
{
    ifstream in("csv.txt");

    if (!in.good())
    {
        cerr << "Error" << endl;
        exit(1);
    }
}
```

```
while (!in.eof())
{
    string line;
    getline(in, line);
    parse(line);
}

return 0;
}
```



```
void parse(string input)
{
    if (regex_match(input, regex( "^$|^#.*$" )))
        return;

    smatch          m;
    ostringstream   sout;

    while (input.length() > 0)
    {
        if (regex_match(input, regex( "[^\\"]*\\\"[^\"]*" )))
        {
            cerr << "Unbalanced \"" << endl;
            return;
        }

        if (regex_search(input, m, regex( "(^(?:\\\"([^\"]+)\\\")|([^\",\\r\\n]*))(?:[,,(?:\\r\\n)]?)" )))
            sout << left << setw(20) << regex_replace(string(m[1]), regex( "\\\"\"" ), "\\\"" );

        input = m.suffix().str();
    }

    cout << sout.str() << endl;
}
```

PARSING CSV INPUT



CSV EXAMPLE REGULAR EXPRESSIONS

- $^{\wedge}\$|^{\wedge}\#.*\$$
- $[^{\wedge}\backslash"]*^{\wedge}\backslash"[^{\wedge}\backslash"]*$
- $(r_1|r_2)r_3$
 - $r_1 = ^{\wedge}(?:^{\wedge}\backslash"([^{\wedge}\backslash"]+)^{\wedge}\backslash")$
 - $r_2 = ([^{\wedge},\\n]*)$
 - $r_3 = (?:[,\\n]?)$
- $(^{\wedge}\backslash"\backslash") \rightarrow ^{\wedge}\backslash"$



RETURNING CSV FIELDS

```
typedef vector<string> field_list;

field_list parse(string input)
{
    field_list fields;

    while(...)
    {
        if (regex_match(input, m, ...)
            fields.push_back(regex_replace(string(m[1]), ...));
        }
    }

    return fields;
```