/// <summary>

/// Leet code #9904. CSV Parser

///

/// The CSV file contains lines of phrase delimited by , when there is

/// a , in the phrase, we use double quote to qoute it, for example

/// "San Francisco, CA", if there is a double quote in the phrase,

/// we use another double quote in front of it to escape it.

///

/// For example, assuming we have following lines

/// John,Smith,john.smith@gmail.com,Los Angeles,1

/// Jane, Roberts, janer@msn.com, "San Francisco, CA", 0

/// "Alexandra ""Alex""", Menendez, alex.menendez@gmail.com, Miami, 1

/// """Alexandra Alex"""

///

/// After we parse it, it become the followng vector of strings.

/// John | Smith | john.smith@gmail.com | Los Angeles | 1

/// Jane | Roberts | janer@msn.com | San Francisco, CA | 0

/// Alexandra "Alex" | Menendez | alex.menendez@gmail.com | Miami | 1

/// "Alexandra Alex"

/// </summary>

/// <summary>

/// Leet code #9904. CSV Parser

/// </summary>

string LeetCode::parsePhrase(string& line, int &pos)

{

string result;

bool in\_quote = false;

while (pos < (int)line.size())

{

// ignore leading space

if (isspace(line[pos]))

{

// but do not ignore the space bwteen words or in quote

if (!result.empty() || in\_quote) result.push\_back(line[pos]);

}

else if (line[pos] == '"')

{

// double quote, it is a quote in phrase

if ((pos < (int)line.size() - 1) && (line[pos + 1] == '"'))

{

result.push\_back('"');

pos++;

}

else

{

// quote to enclose the phrase

in\_quote = !in\_quote;

}

}

else if (line[pos] == ',')

{

if (in\_quote) result.push\_back(',');

else break;

}

else

{

result.push\_back(line[pos]);

}

pos++;

}

// trim right

int index = result.size() - 1;

while (index >= 0 && isspace(result[index])) index--;

result.substr(0, index + 1);

return result;

}

/// <summary>

/// Leet code #9904. CSV Parser

/// </summary>

vector<string> LeetCode::parseLine(string& line)

{

int pos = 0;

vector<string> result;

while (pos < (int)line.size())

{

if (line[pos] == ',')

{

pos++;

}

result.push\_back(parsePhrase(line, pos));

}

return result;

}

vector<vector<string>> LeetCode::parseCSV(vector<string>& lines)

{

vector<vector<string>> result;

for (size\_t i = 0; i < lines.size(); i++)

{

result.push\_back(parseLine(lines[i]));

}

return result;

}