

CPSC 323 - Compilers and Languages

Summer 2017

Programming Assignment 1

In this assignment, you will use regular expressions to match patterns and extract information from an INI file.

Input

Input will be provided in the form of an [INI file](#). As the Wikipedia article notes, the exact details of the INI file format vary across implementations. You are responsible for the following:

- The basic features, as shown in the following sections:
 - [Keys \(properties\)](#)
 - [Sections](#)
 - [Case insensitivity](#)
 - [Comments](#)

In each of these sections, use the details described as Windows-specific.

- Under [Varying features](#), your program should allow for the following:
 - [Blank lines](#) should be allowed
 - Comments need not occur only on lines by themselves
 - [Duplicate names](#) for properties should override previous occurrences.
 - Duplicate names for sections should have their properties merged together
 - [Global properties](#) should be treated as if they belong to a section named `global`
 - Leading and trailing [whitespace](#) around property names should be ignored
 - Trailing whitespace in property values should be ignored
 - Other than global properties, the [order of sections and properties](#) is irrelevant.

Processing

Read the file line-by-line, using [regular expressions](#) and the C++ [std::regex](#) library. Use capturing groups to extract section names, property names, and values.

As elements are extracted, store the properties for each section in an appropriate data structure such as [std::map](#) so that they can be retrieved later. Note that data structures such as unordered arrays which require sequential search are *not* appropriate for such a task.

Output

As each line is processed, output zero or more of the following items as they are found:

SECTION *name*
PROPERTY *name*
VALUE *string*

For example, a file containing

[foo]
bar=baz
qux=quux

should result in the output

SECTION foo
PROPERTY bar
VALUE baz
PROPERTY qux
VALUE quux

API

Expose the functionality of your program for use by other programs through the following class definition:

inifile.h

```
#ifndef INIFILE_H_
#define INIFILE_H_

class IniFile
{
public:
    IniFile(string filename);
    string GetProfileString(string section, string property_name);
};

#endif // INIFILE_H_

```

Grading

This assignment is worth 15 points:

- 1 point for submitting the program
- 1 point for compiling without errors
- 1 point for compiling without warnings (using `gcc -Wall`)
- 2 points for handling basic features of INI files
- 2 points for handling advanced (“varying”) features of INI files
- 2 points for appropriate use of regular expressions and capturing groups
- 2 points for appropriate use of data structures to store and retrieve elements
- 2 points for producing correct output
- 2 points for implementing the API correctly