

PROG71985 – Winter 2019

Coding Assignment #3

The ability to output files that are compatible with other major applications is of tremendous benefit to a programmer. As an example, being able to format a file to hold data collected from a sensor for analysis in a tool like Microsoft Excel is an extremely valuable capability – this interoperability means the programmer need not implement data analysis tools within their custom software, and instead, can merely share the data in a convenient fashion for another application to perform that task.

In this assignment you will practice some basic data file processing techniques, and multi-file project development techniques. Visit the “Tips” folder in eConestoga for details on multi-file projects and the make utility.

Assignment Requirements

Create source file *fileio.c*, where you will develop two functions, known as Application Programming Interfaces (APIs) that will support data interchange with Microsoft Excel. Your application will have a main function that will collect raw data, and provide two means of saving that data in files that are accepted by Excel: CSV (Comma Separated Values) and XML (eXtensible Markup Language).

The first function you will develop matches this function signature:

```
int saveDataInCSV (char time[][9], float values[], int numValues, char filename[]);
```

The purpose of this function is to take an array of date strings (formatted as 9 character rows with this format: HH:MM:SS with a null terminator for a total of 9 characters of storage per row), and associated with each time is a floating point sensor value. The 3rd parameter provides the count of how many rows are in each array, and the 4th parameter provides the filename to save under. The return value indicates success (return value of 0), or a negative number indicating a potential error that has arisen. The file output generated by this function will CSV format, with one column showing time, other showing values.

The second function you will develop matches this function signature:

```
int saveDataInXML (char time[][9], float values[], int numValues, char filename[]);
```

The purpose of this function is to take an array of date strings (formatted as 9 character rows with this format: HH:MM:SS with a null terminator for a total of 9 characters of storage per row), and associated with each time is a floating point sensor value. The 3rd parameter provides the count of how many rows are in each array, and the 4th parameter provides the filename to save under. The return value indicates success (return value of 0), or a negative number indicating a potential error that has arisen. The file output generated by this function will be in XML format, compatible with Microsoft Excel’s XML expectations, with one column showing time, other showing values.

Create a header file *fileio.h* that provides prototypes and other shared declarations needed to use the functionality in this source file. Implement as per industry standards.

Create a 3rd source file called *test.c* will use the header file above, and provide a main() function that will generate a 50 element array of random floating point values between 0.0 and 1.0, along with an array of

50 null terminated time strings in HH:MM:SS format. A floating point value will be generated once every second, so the test of your two functions will require 50 seconds followed by a call to each of the functions. To verify your two functions have output data correctly for Excel, you will load the output files into Excel and you should see a two column table – column A holding the time stamp information and column B holding the associated data.

Deliverables

At the due date, in the drop box provided, submit:

- a) Three C source code files – *fileio.c*, *fileio.h* and *test.c* and a *makefile* that supports the implementation of the two required functions and the test mainline. Ensure source code style is similar to the recommended template provided in week 1. If using Tiny C Compiler, build a *makefile* for your project. If using Visual Studio, have the IDE export the equivalent *makefile* to support command line compilation for your VS project.
- b) One CSV file generated by your software and one XML file generated by your software.
- c) Screen snapshots showing result of Excel loading your output files (one for CSV, one for XML).

Marking Scheme

This assignment will be graded out of 10 points, as follows:

Implementation of CSV output – 2 points:

- 0 = no implementation of CSV output functionality
- 1 = function exists but CSV output not compatible with Excel
- 2 = function exists and CSV output accepted by Excel (proved via screen snapshot)

Implementation of XML output – 2 points:

- 0 = no implementation of XML output functionality
- 1 = function exists but XML output not compatible with Excel
- 2 = function exists and XML output accepted by Excel (proved via screen snapshot)

Use of tools - Multi-source file implementation – 2 points:

- 0 = solution not implemented as a multi-source file project
- 1 = solution separated into the three required files, makefile provided but does not compile cleanly
- 2 = solution separated into the three required files, makefile provided and compiles cleanly

Professionalism in Software Engineering – 2 points:

- 0 = no source code provided to evaluated, or code unable to compile
- 1 = code has been provided, compiles, but not in any discernable style, not using template
- 2 = exemplary code that is documented as expected (header comment blocks, function comment block, in-line comments explaining code, appropriate coding style used throughout)

Correct Implementation – maximum 2 marks deducted for coding errors

- Syntax errors and/or runtime logic errors will be reviewed for solution correctness

APIs to explore:

windows.h: Sleep(), GetLocalTime(), GetTickCount()

stdio.h: sprintf(), fopen(), fclose(), fprintf()

stdlib.h: srand(), rand()

Appendix A – Graduate Attributes

Professional Body of Knowledge

K B	KB 1	KB 2	P A	PA 1	PA 2	PA 3	IV	IV 1	IV 2	IV 3	E D	ED 1	ED 2	ED 3	ED 4	ED 5	E T	ET 1	ET 2	ET 3	ET 4
Knowledge base	Facts	Concepts	Problem analysis	Decomposition	Methodology	Validation	Investigation	Research	Measure	Experiment	Design	Problem Def'n &	Preliminary Design	Detailed Design	Implementation	Verification & Valid'n	Use of engineering tools	Models/Simulations	Measurement Tools	Manufacturing Tools	CAD Systems
	D	D										D A	D	D	D A	D A				I D A	

Employability Skills

TM	TM1	TM2	TM3	CM	CM1	CM2	PR	PR1	PR2	PR3	PR4	LL	LL1	LL2	LL3	LL4
Individual and team work	Personal Contribution	Collaboration	Infrastructure	Communication skills	Log Engineering Info	Convey Engineering Info	Professionalism	Work Ethic	Professional Conduct	Professional Contribution	Professional Practice	Life-long learning	Autonomous Learning	Applying Knowledge & Skills	Self Direction & Reflection	Learning Strategies (Metacognition)
	A					D A		D	D		D					

Professional Responsibility

SC	SC1	SC2	SC3	EE	EE1	PM	PM1	PM2	PM3	PM4	PM5
Impact on society and the environment	Environmental Awareness	Product Life Cycle	Balance & Tradeoff	Ethics and equity	Ethical Responsibility	Economics and project management	Project Scheduling	Resource Allocation & Costing	Risk Management	Business Planning	Economic Analysis
		D									

LEGEND		
I	Introduced	first experience/use
D	Developed	continued experience/use
A	Applied	integration/extension of knowledge & skills