Assignment#1:

FILE TRANSFORMATION SYSTEM

Due date

7th April, 23:55

Goal

In this assignment, you are asked to develop a command line tool to convert files between CSV, Binary, and XML file formats. In addition, to validate your XML file, a corresponding XSD file should be prepared. All codings must be in C programming language. Assignment will be done in groups of three.

This assignment is expected to help you practice basic file operations and understand the details of different file formats, as well as practicing C programming language.

Implementation Details and Requirements

A sample CSV file ("records.csv") is given to you to test your program. This file consists of some information about the students registered in a course.

- The details of the features are as follows:

Attribute Name	Description
name	Displays the name of a student (assuming it as at most 19 characters) – required field
surname	Displays the surname of a student (assuming it as at most 29 characters) – required field
stuID	Displays the student ID that starts with 4 digits, followed by "510", and ends with 3 digits – required field
gender	May have one of the values: "*M*" or "*F*" – required field
email	May have one of the email addresses ending with: "*@gmail.com*","*@hotmail.com*", "*@yahoo.com*", or "*@ogr.deu.edu.tr*"
phone	Displays the phone number in a specific format (Starting with (+90) and continue with 3 digits followed by a dash (-), 3 digits followed by a dash (-), ending with 4 digits, such as "*(+90)504-845-1427*"
letter_grade	Displays the letter grade of the student in the specified course; it can be one of the following letter grades: "AA", "BA", "BB", "CB", "CC", "DC", "DD", or "FF" – required field
midterm	Displays the midterm grade of the student (with the value of minimum 0 and maximum 100)
project	Displays the project grade of the student (with the value of minimum 0 and maximum 100)
final	Displays the final grade of the student (with the value of minimum 0 and maximum 100)
regularStudent	May be one of the emojis: " 👍 ", or " 👎 "
course_surveyRating	Survey rating given by the student for the specified course (between 1 and 5)

- In the first part, you should read the given CSV file and convert it to a binary file.

- The application you will develop will accept some set up parameters from a JSON file, named "setupParams.json". The general format of this file can be as follows:

```
{
    "dataFileName": "records.dat",
    "keyStart": 51,
    "keyEnd": 60,
    "order": "ASC"
}
```

where *dataFileName* indicates which file to be handled in the conversion operations from binary to XML, *keyStart* and *keyEnd* are the starting and ending offsets of the key field in each record that will be used for ordering, and finally *order* is used to organize records in an ascending or descending order considering ASC and DESC values, respectively. For instance, according to the sample json file, records.dat file will be read. After reading the binary file, it is apparent that *keyStart* and *keyEnd* values indicate the field "stuID". Becase the parameter *order* is given as "ASC", all of the data pulled from the records.dat file should be ordered in an ascending order according to the *stuID* field. After handling the parameters in setupParams.json, the data in the correct order should be written to XML file.

- XML file should be in the given format below:

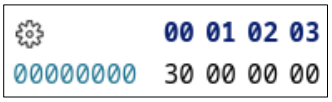
```
<records>
        <row id="1">
                <student_info>
                        <name>James</name>
                        <surname>Butt</surname>
                        <stuID>2021510049</stuID>
                        <gender>M</gender>
                        <email>jbutt@ogr.deu.edu.tr</email>
                        <phone>(+90)504-845-1427</phone>
                </student_info>
                <grade_info letter_grade="CB">
                        <midterm>47</midterm>
                        <project>70</project>
                        <final>74</final>
                </grade_info>
                <regularStudent>👍</regularStudent>
                <course_surveyRating hexVal_bigEnd="00000005" hexVal_littleEnd="05000000" decimal="83886080">5</course_surveyRating>
        </row>

                                                                ...

</records>
```

- Please pay attention that the root element of the output XML file is the name of the output file. For each line that was read from the file, a row number is assigned as the "id" attribute starting from 1 and its value is increased by 1. The tags of the subelements under "row" in the XML file are found at the header of the "*records.csv*" file.
- The field "*course_surveyRating*" is an element comprised of a value and three attributes. The content of this element is a rating value. The first attribute is the corresponding hexadecimal value of the given number in the Big Endian format, second attribute is the corresponding hexadecimal value of the given number in the Little Endian format. The last attribute is the converted value of the number in the attribute *hexVal_littleEnd* to its corresponding decimal value.

To clarify it, in the below scenario, hex editor is little endian. Considering the system is 64 bit, an integer value is placed in the first 4 bytes of a binary file. According to the hexadecimal value, the number is "48" in decimal format. Because, the least significant byte is read first.



If the above hex editor was big endian, the ordering of the reading would change and the most significant byte would be read first. In this case, the value would be read as "805306368" in decimal format.

- The tool takes command line arguments according to the formats you want to convert between them. A typical command line usage is as follows:

```
./myConverter <input_file> <output_file> <type>
```

- `myConverter` is the executable file name, the first argument, `<input_file>` refers to the source file to be used for the conversion and the second one, `<output_file>`, refers to the target file, or XSD file. The last argument, `<type>` defines conversion type (1=CSV to BIN, 2=BIN to XML, 3=XML validation with XSD).

- A sample command line usage converting from binary file to XML is as follows:

```
./myConverter records.csv records.dat 1
```

- A sample command line usage converting from binary file to XML is as follows (please note that no `<input_file>` is declared, because it will be taken from the given in the "setupParams.json" file):

```
./myConverter records.xml 2
```

- You should also **create an XSD file** that will be used to validate your XML. XSD file should include all properties including patterns and restrictions. "validation.c" file is shared with you to test your XSD file on the corresponding XML file. A sample command line usage for XML validation is as follows:

```
./myConverter records.xml recors.xsd 3
```

Documentation

In this assignment, inline documentation is expected, as well as good coding practices such as consistent naming, proper usage of indentation and high readability of code.

Submission

- Name your source code file *x*.c and XSD file *x*.xsd, where *x* is your ***student id***. If you don't follow the naming rules, a penalty applies. (10 pts)
- Late submission is NOT ACCEPTED.

Honesty

Your submissions will be scanned among each other as well as the Internet repository **(including ChatGPT)**. Any assignments that are over the similarity threshold of a system for Detecting Software Similarity will get zero. We strongly encourage you not to submit your assignment rather than a dishonest submission.

Grading policy

- ✓ CSV file reading – 20%
- ✓ JSON parsing – 10%
- ✓ Binary file creation – 10%
- ✓ Binary file reading – 10%
- ✓ XML file creation – 15%
- ✓ XSD file preparation – 15%
- ✓ Little Endian - Big Endian conversion – 10%
- ✓ Inline documentation – 10%

For Questions

For any questions about the assignment please write under the topic "Assignment1 Questions" in Forum on the SAKAI platform. Before asking your question, please check carefully previous questions and answers, where similar questions that were asked by someone else were already answered. No private questions via email will be answered!!!

Good luck!!!