Assignment#1: Bin2XML: File Convertor

Due date: 14th May Friday, 23:59

Goal

In this assignment, you are asked to develop a command line tool to convert a binary file to XML format. All coding must be in C programming language.

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

Implementation Details & Requirements

- We give a sample binary file ("*records.dat*") to test your program. This file consists of some information about the employees of a company. Each record has the following attributes and their sizes are given below:

```
struct record {
    char name[64];          //utf16
    char surname[32];       //utf8
    char gender;
    char email[32];
    char phone_number[16];
    char address[32];
    char level_of_education[8];
    unsigned int income_level;   // given little-endian
    unsigned int expenditure;    // given big-endian
    char currency_unit[16];
    char currentMood[32];
    float height;
    unsigned int weight;

};
```

- The details of the attributes are as follows:

Attribute Name	Description
name	Displays the name of a person
surname	Displays the surname of a person
gender	May have one of the values: "*M*" or "*F*"
email	May one of the email addresses: "*@gmail.com*", "*@hotmail.com*", or "*@yahoo.com*"
phone_number	Displays the phone number
address	Displays the address of the person
level_of_education	May have one of the values "*PhD*", "*MSc*", "*BSc*", "*HS*", or "*PS*" where *PhD*: Doctor of Philosophy, *MSc*: Master of Science, *BSc*: Bachelor of Science, *HS*: High School, *PS*: Primary School

income_level	Displays the income level of the person
expenditure	Displays the expenditure of the person
currency_unit	May have one of the following units: “€”, “₺”, or “$”
current_mood	May be one of the emojis: “😊”, “☹”, “😭”, “😁”
height	Displays the height of the person
weight	Displays the weight of the person

- Bin2XML will **produce an XML file** that looks like the following:

```
<records>
      <row id=”1”>
            <name>James😉</name>
            <surname>Butt</surname>
            <gender>M</gender>
            <email>jbutt@gmail.com</email>
            <phone_number>504-845-1427</phone_number>
            <address>7 W Cerritos Ave #54</address>
            <level_of_education>MSc</level_of_education>
            <income_level bigEnd=1798766592>14187</income_level>
            <expenditure bigEnd=3758686208>2528</expenditure>
            <currency_unit>$</currency_unit>
            <currentMood>😊</currentMood>
            <height>1.33</height>
            <weight>68</weight>
      </row>
      ...
</records>
```

- Please pay attention that the root element of the output XML file is the name of the output file. For each person 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 element names of “row” in the XML file are found at the beginning of the “*records.dat*” file.

- The field “*expenditure*” should be read in the **Big Endian format** whereas the field “*income_level*” should be read in the **Little Endian format**. These fields should be written in XML file as two different versions: the original value read from the file and its converted version to the Big Endian format. For example, the “*income_level*” field is read in the Little Endian format and it should be given as the value of the element. The converted value of “*income_level*” to the Big Endian format should be given as the attribute of the respective element.

- The field “*surname*” should be read in **UTF-8 format** whereas the field “*name*” should be read in **UTF-16 format**.

- Usage of Bin2XML must be as follows:

  `Bin2XML <inputfile> <outputfile>`

- The first argument, `<inputfile>` refers to the source filename to be used for the conversion and the second one, `<outputfile>`, refers to the target XML filename.

- The sample command line usage converting from the binary file to XML as follows:

```
Bin2XML personList.dat personList.xml
```

- 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.

Documentation

In this assignment, in line 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 *xxx*.c, where *xxx* is your ***student id***. If you don't follow the naming rules, a penalty applies. (10 pts)
- Late submission is accepted but, 10 points penalty applies for each day.

Honesty

Your submissions will be scanned among each other as well as the Internet repository. 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

- Binary file reading - %25
- XML file creation - %25
- Little Endian & Big Endian reading - %15
- UTF-8 & UTF-16 reading - %15
- XSD validation - %20

For Questions

For any questions about the assignment please write under the topic "Homework1 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!!!
- We will try to answer any of your questions as soon as possible, except the ones "Hocam my code does not work, can you fix it" or "I have implemented it but it does not work, can you look at it". Debuggers are far more suitable options.

Good luck!!!

Read all of the instructions carefully, if you find something UNCLEAR, please ask help to CLARIFY it!