DOKUZ EYLUL UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING

CME 2202 DATA ORGNIZATION and MANAGEMENT

ASSIGNMENT-1

File Convertor - Bin2XML

Lecturers

DR.ÖĞR.ÜYESİ ÖZLEM AKTAŞ

DR. MELTEM YILDIRIM EKİCİ

DR. İBRAHİM ATAKAN KUBİLAY

YASSER EL HASAN

2019510006

İZMİR

24/04/2022

1. Introduction :

In this assignment, I have developed a command-line tool to convert a binary file to XML format. Where the program takes the path of the source and the destination file to perform the reading and writing operations.

2. Algorithms and Solutions :

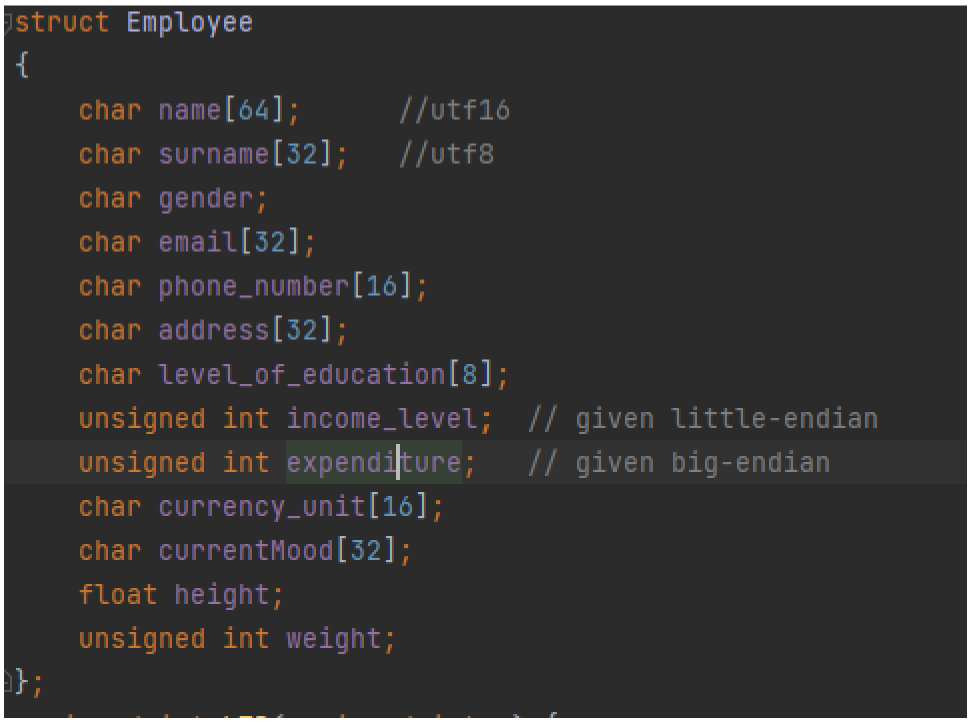
Basically, The project is done using the C language. It consists of one main class that contains the necessary methods and structs. there is the **ReadEmployee** method that reads the records of employees from a (dat) file, and fills the structs by information read from the records file depending on the variables in the structs. Where it makes many of copies of employees with different data.

Besides, there are **BTL** and **LTB** (little and big-endian) methods that perform the converting from little to big-endian and from big to little-endian.
In the **main** method, after the reading operation is done, I start creating the XML file, where I construct a file that will be filled by the employees' structs, with controlling the writing operation to get a correct form of XML file, where I consider the tags and the other syntax rules of XML

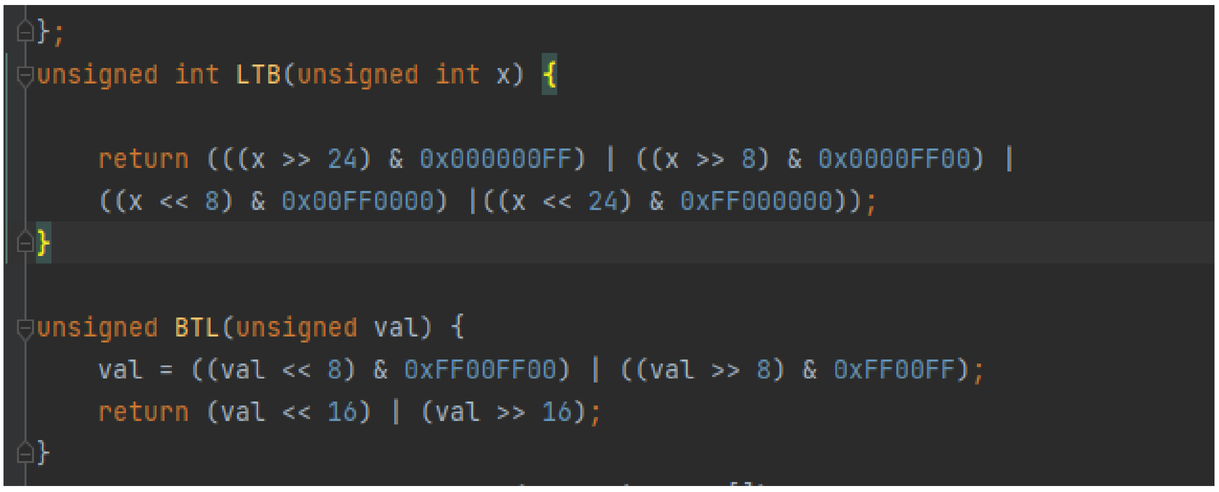
3. Problems Encountered:

As a main problem that I encountered was validating the XML file using XSD file, where I did a lot of research about this problem, but I could not find anything helps me to perform this task, that I faced a problem while reading and writing in different codings like UTF-8 and UTF-16

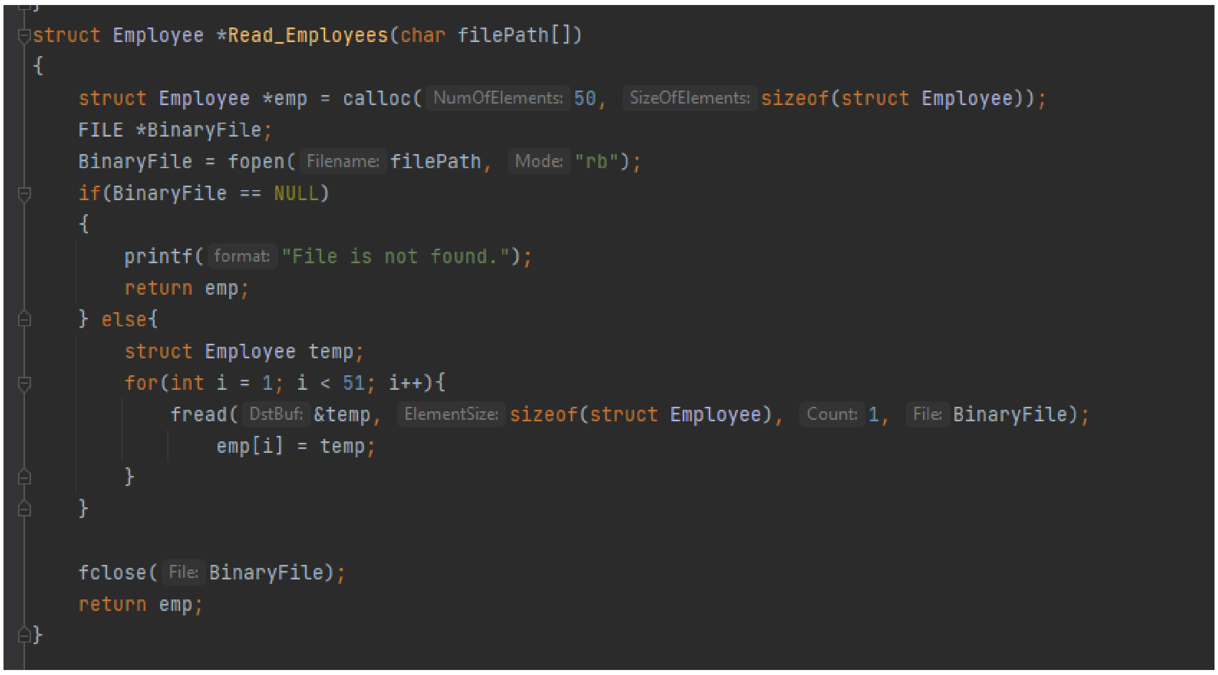
4. Sample Screenshots:



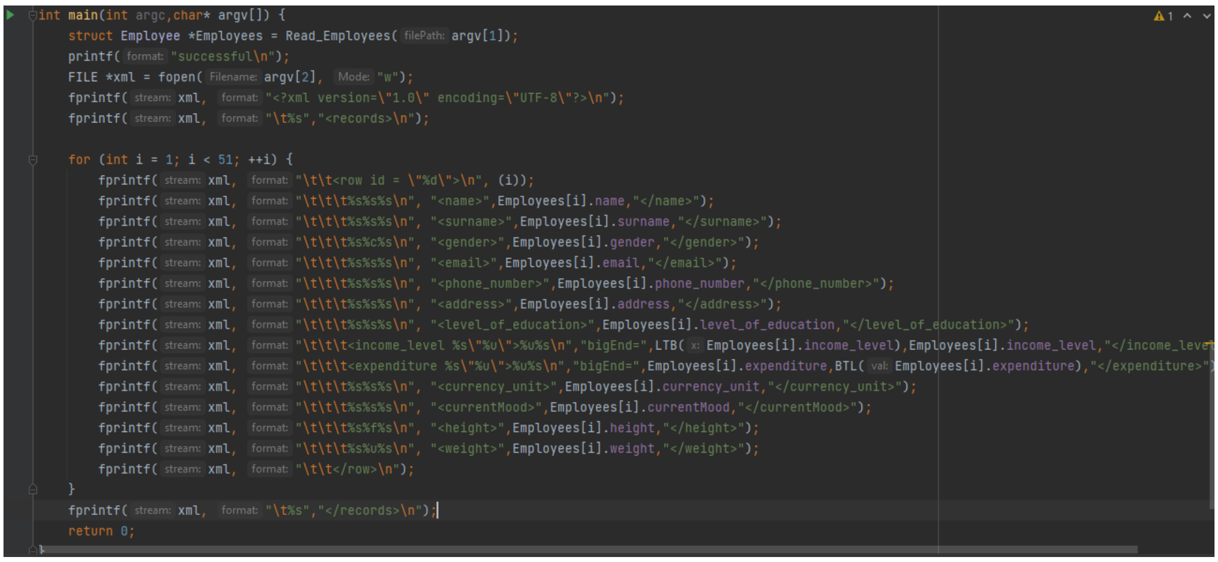
```
struct Employee
{
    char name[64];      //utf16
    char surname[32];   //utf8
    char gender;
    char email[32];
    char phone_number[16];
    char address[32];
    char level_of_education[8];
    unsigned int income_level;  // given little-endian
    unsigned int expenditure;   // given big-endian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



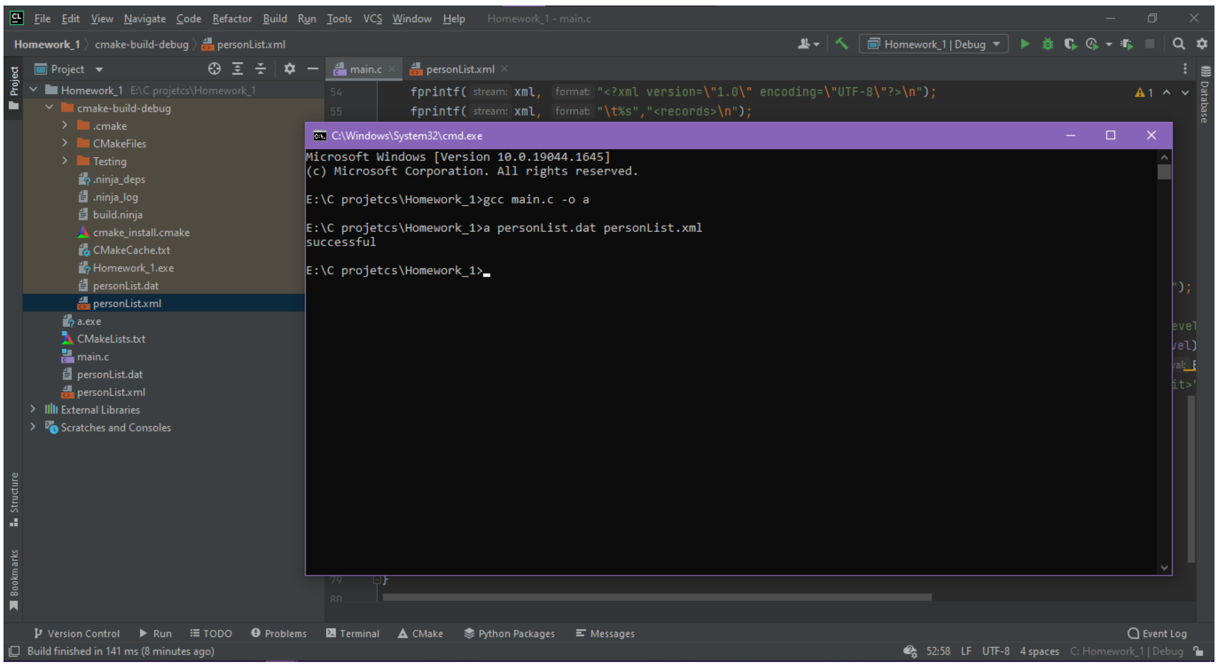
```
};
unsigned int LTB(unsigned int x) {

    return (((x >> 24) & 0x000000FF) | ((x >> 8) & 0x0000FF00) |
    ((x << 8) & 0x00FF0000) |((x << 24) & 0xFF000000));
}

unsigned BTL(unsigned val) {
    val = ((val << 8) & 0xFF00FF00) | ((val >> 8) & 0xFF00FF);
    return (val << 16) | (val >> 16);
}
```



```c
struct Employee *Read_Employees(char filePath[])
{
    struct Employee *emp = calloc( NumOfElements: 50,  SizeOfElements: sizeof(struct Employee));
    FILE *BinaryFile;
    BinaryFile = fopen( Filename: filePath,  Mode: "rb");
    if(BinaryFile == NULL)
    {
        printf( format: "File is not found.");
        return emp;
    } else{
        struct Employee temp;
        for(int i = 1; i < 51; i++){
            fread( DstBuf: &temp,  ElementSize: sizeof(struct Employee),  Count: 1,  File: BinaryFile);
            emp[i] = temp;
        }
    }

    fclose( File: BinaryFile);
    return emp;
}
```

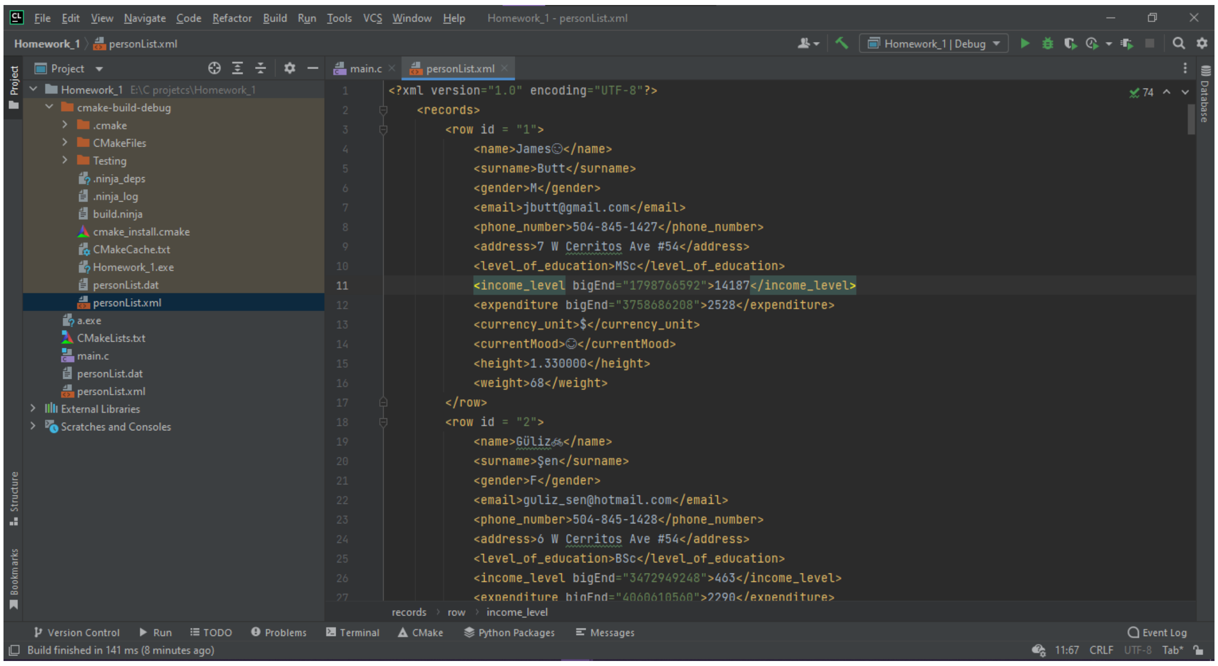


```c
int main(int argc,char* argv[]) {
    struct Employee *Employees = Read_Employees( filePath: argv[1]);
    printf( format: "successful\n");
    FILE *xml = fopen( Filename: argv[2],  Mode: "w");
    fprintf( stream: xml,  format: "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");
    fprintf( stream: xml,  format: "\t%s","<records>\n");

    for (int i = 1; i < 51; ++i) {
        fprintf( stream: xml,  format: "\t\t<row id = \"%d\">\n", (i));
        fprintf( stream: xml,  format: "\t\t\t%s%s%s\n", "<name>",Employees[i].name,"</name>");
        fprintf( stream: xml,  format: "\t\t\t%s%s%s\n", "<surname>",Employees[i].surname,"</surname>");
        fprintf( stream: xml,  format: "\t\t\t%s%c%s\n", "<gender>",Employees[i].gender,"</gender>");
        fprintf( stream: xml,  format: "\t\t\t%s%s%s\n", "<email>",Employees[i].email,"</email>");
        fprintf( stream: xml,  format: "\t\t\t%s%s%s\n", "<phone_number>",Employees[i].phone_number,"</phone_number>");
        fprintf( stream: xml,  format: "\t\t\t%s%s%s\n", "<address>",Employees[i].address,"</address>");
        fprintf( stream: xml,  format: "\t\t\t%s%s%s\n", "<level_of_education>",Employees[i].level_of_education,"</level_of_education>");
        fprintf( stream: xml,  format: "\t\t\t<income_level %s\"%u\">%u%s\n","bigEnd=",LTB( x: Employees[i].income_level),Employees[i].income_level,"</income_leve
        fprintf( stream: xml,  format: "\t\t\t<expenditure %s\"%u\">%u%s\n","bigEnd=",Employees[i].expenditure,BTL( val: Employees[i].expenditure),"</expenditure>")
        fprintf( stream: xml,  format: "\t\t\t%s%s%s\n", "<currency_unit>",Employees[i].currency_unit,"</currency_unit>");
        fprintf( stream: xml,  format: "\t\t\t%s%s%s\n", "<currentMood>",Employees[i].currentMood,"</currentMood>");
        fprintf( stream: xml,  format: "\t\t\t%s%f%s\n", "<height>",Employees[i].height,"</height>");
        fprintf( stream: xml,  format: "\t\t\t%s%u%s\n", "<weight>",Employees[i].weight,"</weight>");
        fprintf( stream: xml,  format: "\t\t</row>\n");
    }
    fprintf( stream: xml,  format: "\t%s","</records>\n");
    return 0;
}
```



```
fprintf( stream: xml, format: "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n");
fprintf( stream: xml, format: "\t%s","<records>\n");
```

```
Microsoft Windows [Version 10.0.19044.1645]
(c) Microsoft Corporation. All rights reserved.

E:\C projetcs\Homework_1>gcc main.c -o a

E:\C projetcs\Homework_1>a personList.dat personList.xml
successful

E:\C projetcs\Homework_1>
```



```
<?xml version="1.0" encoding="UTF-8"?>
    <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>1.330000</height>
            <weight>68</weight>
        </row>
        <row id = "2">
            <name>Güliz🚲</name>
            <surname>Şen</surname>
            <gender>F</gender>
            <email>guliz_sen@hotmail.com</email>
            <phone_number>504-845-1428</phone_number>
            <address>6 W Cerritos Ave #54</address>
            <level_of_education>BSc</level_of_education>
            <income_level bigEnd="3472949248">463</income_level>
            <expenditure bigEnd="4060610560">2290</expenditure>
```