Practical Work – FSDS: "LOF Data File with a Dense Main-Memory Index and a TOF Index File" – Programming in C

We assume that we want to create and manage a binary file named "**STUDENTS_ESI.BIN**" containing all students enrolled at ESI across the different years of study: 1CP, 2CP, 1CS, 2CS, and 3CS. This file will be organized using the "**LŌF** or **LnOF**" method, meaning it is structured as a list of blocks containing fixed-size records that are not ordered. Each block may contain up to 40 records.

☞ Required Work

1. Program and use the abstract machine model presented in class. As a reminder, this model includes the following functions: *ReadBlock, WriteBlock, getHEADER, setHEADER, Open, Close,* and *AllocBlock*.

2. Program a module to create the file STUDENTS_ESI.BIN, where each record consists of 10 fields: < Student_ID, Family_Name, First_Name, Date_Birth, Wilaya_Birth, Gender, Blood_Type, Year_Study, Speciality, Resident_UC >. The creation of this file will rely on an initial loading module, which fills the file with N records (N being an integer to be read, for example: 1500, 2000, …).
Calculate and display the cost of this creation operation, denoted as C2.

For each newly inserted record, its identifier (key) and its coordinates (block number and offset) are inserted into an index table (in main memory) with shifts to maintain the order of the keys.

To ensure that the initial loading of STUDENTS_ESI.BIN is fast, it should be performed using random values as follows:

- **Student_ID (record key):** Generate a random number between 1000 and 9000.
- **Family_Name / First_Name:** First, generate a number between 4 and 30 (representing the number of characters in the name). Then, for each character of the name, generate a number between 1 and 26 and assign the corresponding letter.
- **Date_Birth:** Generate a random number between 2003 and 2008 for the year, a random number between 1 and 12 for the month, and a random number between 1 and the number of days in the generated month (28, 29, 30, or 31) for the day. All date validations must be respected (e.g., leap year considerations, etc.).
- **Wilaya_Birth:** Generate a random number between 1 and 58. Then, retrieve the corresponding wilaya name based on this number (e.g., 16 : Algiers, 23 : Annaba, etc.).
- **Gender:** Generate a random number: 1 for male and 2 for female.
- **Blood_Type:** Generate a random number between 1 and 8. Then, assign the corresponding blood type (1: O+, 2 : A+, 3 : B+, 4 : O-, 5 : A-, 6 : AB+, 7 : B-, 8 : AB-).
- **Year_Study:** Generate a random number between 1 and 5. This number directly determines the student's year of study as follows: 1 : 1CP, 2 : 2CP, 3 : 1CS, 4 : 2CS, 5 : 3CS.
- **Speciality:** This field depends on the value of *Year_Study*. If the year is 1CP or 2CP, the speciality is "Integrated Preparatory Classes"**.** If the year is 1CS, the speciality is "Common Core". If the year is 2CS or 3CS, generate a random number between 1 and 4 to determine the speciality: (1 : Information Systems and Technologies (SIT), 2 : Computer Systems (SIQ), 3 : Software and Computer Systems (SIL), 4 : Intelligent Systems and Data (SID)).
- **Resident_UC:** This field, abbreviation for *Resident at the University Campus*, is a boolean indicating whether the student resides in the university campus.

3. Program the modules listed below (displaying the cost of each operation):

   3.1. Save the index to a TOF index file named "StudentID_INDEX.idx" (Cost C31)

   3.2. Load the index into main memory from the file StudentID_INDEX.idx (Cost C32)

   3.3. Search for a student by their identifier. This module must return the block address as well as the position within the block. (Cost C33)

   3.4. Insert a new student into the file STUDENTS_ESI.BIN. The identifier must be generated randomly and be unique (no duplicates). (Cost C34)

   3.5. Delete a student given their identifier. (Cost C35)

   3.6. Modify the first name of a given student (Last_Name). (Cost C36)

4. Program the modules listed below (displaying the cost of each operation). To speed up processing, you may use other appropriate indexes in main memory and files to store these indexes.

   4.1. Display all students with a given blood type who reside in the university campus. (Cost C41)

   4.2. Display all students in a given speciality. (Cost C42)

   4.3. Display all students under 20 years old whose birth years fall within the interval [Y1, Y2]. (Cost C43)

   4.4. Display all records corresponding to a given year of study. (Cost C44)

5. From the file STUDENTS_ESI.BIN, create a LOF file named STUDENTS_CP.BIN containing all students enrolled in 1CP and 2CP. The load factor must be 75%. (Cost C5).

6. Program the following optional modules:

   6.1. Display the header of a given file (STUDENTS_ESI.BIN or STUDENTS_CP.BIN).

   6.2. Display the contents of a given block from a given file.

   6.3. Display the contents of the entire file block by block for a given file.

☞ Important points :

- All requested modules must be executed through a main menu.
- You must submit your source codes and the generated data and index files by uploading them to my Drive via an online form, which will be provided to you shortly. The final deadline for submission of the PW is **Monday, January 5, 2026, before 11:59 PM**. No late submissions will be accepted.
- You must follow the following naming format for your files: **LASTNAME1_LASTNAME2_TP2_Gi.zip**, where *i* represents your group number (compressed file containing only your source codes, data files, and index files used). Failure to follow this format will result in penalties.
- Several aspects will be considered when evaluating your PW, including the demonstration, the source code (structures, comments, etc.), the user interface (UI), and other relevant criteria.

Good luck ! بالـتوفيــق