

# Department of Information Systems and Technologies

## CTIS 152 – Data Structures and Algorithms

Fall 2024 - 2025

### Lab Guide #14 – Week 10-1

**OBJECTIVE :** Recursive Functions and Binary Files.

**Instructor :** Burcu LIMAN

**Assistants :** Berk ÖNDER - Engin Zafer KIRAÇBEDEL

**Q1.** In a library, there are three book shelves and 10 books can be placed in each shelf. All shelves in the library have **shelf numbers**, **indexes** formed by using names and ISBNs of the books, the **number of books** and **the list of books** in each shelf. Each **book** has **book name**, **ISBN**, **author** and **publish year**.

Write a C program that reads the books' information for 3 shelves from an unsorted file named "**books.txt**" into a nested structure array. Each line of the file contains; shelf number, book name, isbn, author and publish year of the book.

The program sorts the books in all shelves in ascending order, then creates indexes for each shelf by using the functions below and displays them on the screen.

Write the following functions;

- **bubbleRec** that sorts the books on each shelf by book name in ascending order using the **bubble sort** algorithm recursively.
- **makeIndex** that makes indexes of each shelf. For example; if the first book's name is "**Conceptual Science**", the last book's name is "**Science and Technology**": program takes C and S, then the ISBNs' first part (till -) and combines them as "**C-S; 1110-3452**" for the index of a shelf.

#### books.txt

```
2 Problem Solving 1234-SC28-ML69 Walter Savitch 2006
1 Science and Technology 3452-LK11-OL90 Marc Paye 2009
3 English Literature 1627-JP18-22UL Jay Stevenson 2007
2 Design Patterns 1978-UL62-22PL Ralph Johnson 1994
3 Norton Anthology 1393-MS90-A256 Sarah Lawall 2005
1 Conceptual Science 1110-ZE29-231E John Suchocki 2007
1 Earth Science 2613-YU27-32RE Dennis Tasa 2008
2 Agile Softwares 1445-KL22-3559 Robert Martin 2002
2 Software Engineering 1071-CA11-4031 Roger Pressman 2004
```

**Project Name:** LG14\_Q1

**File Name:** Q1.cpp

#### Example Run:

```
Contents of ALL Shelves
*****
Shelf No: 1
Conceptual Science          1110-ZE29-231E
Earth Science              2613-YU27-32RE
Science and Technology     3452-LK11-OL90
Index of 1. Shelf: C-S;1110-3452
-----
Shelf No: 2
Agile Softwares            1445-KL22-3559
Design Patterns           1978-UL62-22PL
Problem Solving            1234-SC28-ML69
Software Engineering       1071-CA11-4031
Index of 2. Shelf: A-S;1445-1071
-----
Shelf No: 3
English Literature         1627-JP18-22UL
Norton Anthology          1393-MS90-A256
Index of 3. Shelf: E-N;1627-1393
-----
```

- `fwrite(ptr, size, n, outputfileptr);`  
// writes n items of data of the size size into output file starting from ptr.
- `fread(ptr, size, n, inputfileptr);`  
// reads at most n items of data of the size size from the input file into the memory starting from the address ptr.
- `rewind (filepointer);`  
// causes a file pointer to be repositioned to the beginning of the file.
- `fseek(filepointer, offset, whence);`  
// repositions the file pointer to a new position that is offset bytes from the file location given by whence. (SEEK\_SET, SEEK\_CUR, SEEK\_END)
- `int ftell(file pointer);`  
// returns the current file position of the given stream.

## Q2.

- a) Write a C program that reads the binary file named “Q2.bin” for whole numbers for the amount ( num ) specified by the user. The program should store the first *num* amount of numbers in an array and then display the given array on screen, as shown in the example run. Note that there are 500 numbers in the file given, so create an array of the necessary size to be able to store it all if necessary (preferably using dynamic memory allocation).

**Note:** You must read the numbers one by one from the file for this question.

### Example Run:

```
Please enter the number of items to be read: 12
1. 2080
2. 1307
3. 7736
4. 3316
5. 6310
6. 3647
7. 803
8. 3211
9. 1528
10. 1624
11. 1602
12. 9331
```

**Project Name:** LG14\_Q2a

**File Name:** Q2a.cpp

- b) Modify your code from Q2a in such way that the program no longer reads the data from the file using a loop / one by one. Solve the question without any usage of a loop. The example run will be the exact same.

**Project Name:** LG14\_Q2b

**File Name:** Q2.cpp

- c) Modify your code from Q2b in such way that the program no longer reads the file contents as a whole, or in a given amount of numbers, or randomly, in any way. Your program in this part must request a user input for which number out of the 500 in the file should be read from the beginning ( for instance, the 3rd number ) and the program should read solely that number without using any array or a loop. Examine the example run carefully.

### Example Run:

```
Please enter the order of the item to be read: 3
3. 7736
```

**Project Name:** LG14\_Q2c

**File Name:** Q2c.cpp

- d) Modify your code from Q2c in such a way that the program reads the 4th item from the end and then 8th item from the end using its current position.

### Example Run:

```
497. 6320
493. 6701
```

**Project Name:** LG14\_Q2d

**File Name:** Q2d.cpp

## Additional Questions

**AQ1.** Write a C program that gets a phone brand from the user, and searches it through the sorted binary file named “phones.bin” by using **recursive binary search** algorithm.

- If the searched brand is found, the related information and also the number of steps performed will be displayed as in the example run.
- Information of a phone contains **phone brand (with size of 10)**, **stock** and **stock arrive that will arrive** to the market.

Write the following **recursive** function;

- **rBinarySearch**: that searches a **phone brand** through the binary file **not the structure array**.

### Example Run #1:

```
Enter a phone brand to search: Iphone
After 4 step(s) Iphone found with 60 stocks and 75 stocks will arrive
```

### Example Run #2:

```
Enter a phone brand to search: Panasonic
After 4 step(s) Panasonic could not be found!
```

### Example Run #3:

```
Enter a phone brand to search: Xiaomi
After 4 step(s) Xiaomi found with 32 stocks and 80 stocks will arrive
```

### phones.bin

```
Asus 3 10
HTC 6 10
Huawei 28 45
Iphone 60 75
LG 21 30
Nokia 3 15
Oppo 25 50
Sony 2 15
Samsung 17 49
Xiaomi 32 80
```

**AQ2.** Write a C program that will firstly convert the given “input.txt” file into a binary file named “binary.bin”. Then read the binary file as input file and perform the commands to display the letters. The records are composed of (**S for Set** , **C for Current** , **E for End**), **record number** and **a letter**.

Write the functions;

- **writeToBin**: takes the **input** and **output file names**, reads the content of the input text file and writes them to the output binary file.
- **move**: takes **binary input file** and **one record** as parameter, and according to the record’s direction (**S : Set** , **C : Current**, **E : End**), moves to the specified position and display the record information.

For example;

- if the read line is “**S 11 C**” –first, display the letter “C” on the screen and then Set the cursor to 11th record from **TOP**.
- if the read line is “**C -2 F**” – first, display the letter “F” on the screen and then go **2 record up** from the Current record.
- if the read line is “**E 1 Z**” – first, display the letter “Z” on the screen and then go **1 record up** from the Bottom of the file.

In the main program, after converting the text file to binary by using **writeToBin** function, start to read the binary file until the end of file is reached and perform the commands written on the record by using move function.

**HINT:** The program will also check the commands while performing them. If the command is “**E 0 T**” this means that the last record to display.

### If the given txt is as follows:

```
C 5 H
E 3 F
S 2 Q
S 7 L
S 3 W
C -2 E
S 11 L
E 5 G
S 3 T
C -1 A
E 0 O
```

### Example Run will be:

```
HELLO
```