

Department of Information Systems and Technologies

CTIS152 – Algorithms and Data Structures

Spring 2024 - 2025

Lab Guide #14 – Week 9 – 1

OBJECTIVE : Binary Files and Recursive functions

Instructor : Serpil TIN

Assistants : Berk ÖNDER & Hatice Zehra YILMAZ

- `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.

Q1.

- a) Write a C program that reads and displays the numbers in the binary file named “**nums.bin**”, and the count of numbers on the screen.

Project Name: LG14_Q1a

File Name: Q1a.cpp

Example Run:

193 173 172 257 61 781 693 523 153 62 282 169 691 273 389 942 532 753 493 533 608 883 701 90 886
There are 25 numbers in the files

- b) Modify the program **Q1a.cpp** so that the program finds the number of records in the binary file named “**nums.bin**”, creates a dynamic array, and reads the numbers from the file into the array. Then, it displays the array of content on the screen.

Hint: use **fseek** and **ftell** functions to find the number of records in the file.

Project Name: LG14_Q1b

File Name: Q1b.cpp

Example Run:

There are 25 numbers in the files
193 173 172 257 61 781 693 523 153 62 282 169 691 273 389 942 532 753 493 533 608 883 701 90 886

Q2.

a) Write the following function;

- **createBinaryFile** that creates a binary file named **"numbers.bin"** storing 250 numbers generated randomly between 1 - 5000.

Write a C program that reads the binary file named **"numbers.bin"** created by the user 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 the screen, as in the example run. Note that there are 250 numbers in the file given, so create an array of the necessary size 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.

GENERATION OF RANDOM NUMBERS FOR THE BINARY FILE:

1. Use **stdlib.h** (for srand function)
2. Use **time.h** (for time function).
3. **srand(time(0));** for getting different number every time you run the program.
4. For getting a random number between 0 – 50: `num = rand() % 51;`

Project Name: LG14_Q2a

File Name: Q2a.cpp

Example Run:

```
Please enter the number of items to be read: 8
1. 1941
2. 4928
3. 2063
4. 2963
5. 807
6. 3111
7. 3110
8. 2660
```

b) Modify your code from Q2a in such a way that the program no longer uses a loop while writing or reading the data.

NOTE: First, store the random numbers in an array then write the whole content of the array into the binary file.

Project Name: LG14_Q2b

File Name: Q2b.cpp

Example Run:

```
Please enter the number of items to be read: 5
1. 2150
2. 4698
3. 3490
4. 2919
5. 580
```

You may use the given **numbers.bin** file for part c and d.

c) Modify your code from Q2b in such a 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 **250** in the file should be read from the beginning (for instance, the 13th number) and the program should read solely that number without using any array or a loop. Make a data validation for the order number (1-250). Examine the example run carefully.

Project Name: LG14_Q2c

File Name: Q2c.cpp

Example Run:

```
Please enter the order of the item to be read: 251
Please enter the order of the item to be read: -5
Please enter the order of the item to be read: 0
Please enter the order of the item to be read: 13

13. 6420
```

d) Modify your code from Q2c in such a way that the program gets 2 numbers from the user **m** and **n**, reads the mth item from the end, and then the nth item back from its current position.

Project Name: LG14_Q2d

File Name: Q2d.cpp

Example Run:

```
Please enter the order of the item to be read from the end : 5
246. 5916

Please enter the second order of the item to be read from its current position: 11
236. 1136
```

Q3.

- a) The list of surgeries is given in a binary file. Write a C program that will read information about the surgeries from the file named **"surgery.bin"** into the array of structure (**char** doctorName[20], **char** patientName[20], **char** operation[25], **int** time, **char** status[15]). The program displays the number of records in the file by using **fseek** and **ftell** functions. Generates a random number between 1 to n and then, shows this number of surgeries information as in the example run.

Project Name: LG14_Q3a

File Name: Q3a.cpp

Example Run #1:

There are 5 surgery in the file

1 surgery will be displayed

Doctor	Patient	Operation	Time	Status
-----	-----	-----	-----	-----
Tony Stark	Bruce Wayne	heart surgery	2	Deceased

Example Run #2:

There are 5 surgery in the file

5 surgery will be displayed

Doctor	Patient	Operation	Time	Status
-----	-----	-----	-----	-----
Tony Stark	Bruce Wayne	heart surgery	2	Deceased
Natasha Romanoff	Ivan Vanko	kidney surgery	1	Finished
Bruce Banner	Justin Hammer	eye surgery	1	Finished
Peter Parker	Kate Kane	plastic surgery	3	In Progress
Steve Rogers	Thor	brain surgery	7	In Progress

- b) The program will read the **"surgery.bin"** and write surgery information if the status is **"Finished"** into a binary file named **"output.bin"**. Display the output file on the screen.

Project Name: LG14_Q3b

File Name: Q3b.cpp

Example Run :

BINARY FILE CONTENT

Doctor	Patient	Operaiton	Time	Status
-----	-----	-----	-----	-----
Natasha Romanoff	Ivan Vanko	kidney surgery	1	Finished
Bruce Banner	Justin Hammer	eye surgery	1	Finished

- Q4. Use the given structure and the binary file **"surgery.bin"** in Q1 (surgeries are given in ascending order by patient name). Write a C program that will read the binary file and perform binary search operations on the file without using any array.

Write the following function;

- **binSearch:** takes a file pointer and a string to search, and searches the given name recursively without using an array.

Project Name: LG14_Q4

File Name: Q4.cpp

Example Run:

Enter a name to search (END to stop): Ivan Vanko

Doctor	Patient	Operation	Time	Status
-----	-----	-----	-----	-----
Natasha Romanoff	Ivan Vanko	kidney surgery	1	Finished

Enter a name to search (END to stop): Thor

Doctor	Patient	Operation	Time	Status
-----	-----	-----	-----	-----
Steve Rogers	Thor	brain surgery	7	In Progress

Enter a name to search (END to stop): Spiderman
There is NO patient with the name Spiderman in surgery!

Enter a name to search (END to stop): END

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 a **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 about a phone contains **phone brand (with a 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**.

Project Name: LG14_AQ1

File Name: AQ1.cpp

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

<u>phones.bin</u>		
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 first convert the given “**input.txt**” file into a binary file named “**binary.bin**”. Then read the binary file as an input file and perform the commands to display the letters. The records are composed of (**S for Set , C for Current, E for End**), a **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 a **binary input file** and **one record** as a parameter, and according to the record’s direction (**S : Set , C : Current, E : End**), moves to the specified position and displays 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 records 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 the **writeToBin** function, start to read the binary file until the end of the file is reached and perform the commands written on the record by using the 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.

Project Name: LG14_AQ2

File Name: AQ2.cpp

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