Department of Information Systems and Technologies

CTIS151 – Introduction to Programming
SPRING 2023 - 2024

Lab Guide #14 – Week 11 – 2

OBJECTIVES : File Operations and One-Dimensional Arrays

Instructors : Burcu LİMAN
Assistants : Sıla YAPICI, Engin Z. KIRAÇBEDEL

Q1.

a) Write a C program in which you declare an integer data type array with a constant size of 7, and initialize it by using the values given to you below, in order to display the array contents on the screen as shown in the example run.

```
int arr[7] = { 78, 43, 1, 12, 90, 34, 55 };
```

Project Name: LG14_Q1a
File Name: Q1a.cpp

Example Run:

```
The array contents are: 78 43 1 12 90 34 55
```

b) Modify your code from Q1a in such way that the program no longer initializes the array with the values given to you, instead with those given by the user, in order to display the array contents on the screen.

Example Run:

```
Enter 7 numbers: 64 365 783 3 41 7 90
The array contents are: 64 365 783 3 41 7 90
```

Project Name: LG14_Q1b
File Name: Q1b.cpp

c) Modify your code from Q1b in such way that the program no longer gets the values of the array from the user, instead from a text file named values.txt, in order to display the array contents on screen.

values.txt

```
57 21 6 13 145 211 2
```

Project Name: LG14_Q1c
File Name: Q1c.cpp

Example Run:

```
Reading contents of the file, the 7 numbers are: 57 21 6 13 145 211 2
The array contents are: 57 21 6 13 145 211 2
```

Q2. Write a C program that declares an integer array with the **maximum** size **100**, then reads an **even** number of elements (***n***) from the user, reads the elements into the array and swaps adjacent elements using the following **swap** function. The program displays the final form of the array. Make a data validation for even number of elements!!!

Write the following function;

- **swap**: takes two integer number and exchanges their values.

Project Name: LG14_Q2
File Name: Q2.cpp

Example Run:

```
Enter the number of elements: 7
The number of elements should be EVEN.

Enter the number of elements: 3
The number of elements should be EVEN.

Enter the number of elements: 6

Enter array elements:
Enter element on index [0]: 11
Enter element on index [1]: 43
Enter element on index [2]: 78
Enter element on index [3]: 3
Enter element on index [4]: 42
Enter element on index [5]: 83

Array elements after swapping adjacent elements:
43 11 3 78 83 42
```

Q3.

a) Write a program that **initializes** an ID array (111,222,333,444,555,666,777,888,999,100) representing the IDs of employees'.The programgets the salaries of these 10 employees from the user and finds the average of the employee salaries. Then it displays;

- the average salary
- the salaries which are more than the average salary
- the salaries which are less than the average salary
- the number of salaries which are equal to average salary.
- The program should give warning messages if it could not find any salaries greater than or less than the average salary.

Project Name: LG16_Q3a
File Name: Q3a.cpp

Example Run:

```
Enter 10 employees' salaries:
id = 111 salary = 29500
id = 222 salary = 25000
id = 333 salary = 32000
id = 444 salary = 31700
id = 555 salary = 43000
id = 666 salary = 28000
id = 777 salary = 29600
id = 888 salary = 35000
id = 999 salary = 37000
id = 100 salary = 40300

The average salary is 33110.00
Number of salaries equal to average salary is 0

Greater
555 43000.00
888 35000.00
999 37000.00
100 40300.00

Lower
111 29500.00
222 25000.00
333 32000.00
444 31700.00
666 28000.00
777 29600.00
```

b) Solve part (**a**) again, by taking the inputs from the **employee.txt file** and find the average of the employee salaries. Then it writes;

- the salaries which are more than the average salary to **greater.txt** file
- the salaries which are less than the average salary to **lower.txt** file

Display the number of salaries which are equal to average salary.

Project Name: LG14_Q3b
File Name: Q3b.cpp

Employee.txt

```
111 29500
222 25000
333 32000
444 31700
555 43000
666 28000
777 29600
888 35000
999 37000
100 40300
```

greater.txt

```
555 43000.00
888 35000.00
999 37000.00
100 40300.00
```

lower.txt

```
111 29500.00
222 25000.00
333 32000.00
444 31700.00
666 28000.00
777 600.00
```

Q4. Write a C program that reads a sentence from a text file named **sentence.txt** and writes:

a) The reverse of the sentence into another file named **reverseA.txt**.

sentence.txt

```
My brother takes out the trash at exactly 3 pm everyday
```

reverseA.txt:

```
yadyreve mp 3 yltcaxe ta hsart eht tuo sekat rehtorb yM
```

Project Name: LG14_Q4a
File Name: Q2a.cpp

b) The reverse of each word into another file named **reverseB.txt**. (Hint: Words are separated with blanks.)

reverseB.txt:

```
yM rehtorb sekat tuo eht hsart ta yltcaxe 3 mp yadyreve
```

Project Name: LG14_Q4b
File Name: Q2b.cpp

ADDITIONAL QUESTIONS

AQ1.

Write a C program for a word game where there are two players and each player has to write a word on their turn. Player-1 has to start the game by typing a word that starts with 'a', if it doesn't then player-2 wins. If Player-1 starts correctly, then, player-2 must write a word that starts with the last letter of the previous word, and so on.

- For example, Player-1 starts the game by typing a word that starts with 'a', like admire. Since the last letter of that word is 'e', Player-2 should continue with a word that starts with 'e', like education. So on...

Notice: Each word may have different number of letters, therefore, in order to determine the word ending, players must type a dot character '.' right after the word.

Example Run #1:

```
Player-1, enter a word: admire.
Player-2, enter a word: education.
Player-1, enter a word: nose.
Player-2, enter a word: elevator.
Player-1, enter a word: glass.

glass does not start with r!
Game over: Player-2 wins.
```

Example Run #2:

```
Player-1, enter a word: zucchini.

zucchini does not start with a!
Game over: Player-2 wins.
```

AQ2.

Write a modular C program that reads the text file named **"numbers.txt"** and stores them in an array using a function named **readFile(...)** that takes in the file stream and the array as parameters in order to read the data from the text file into the array. Then, write a function named **menu()** that displays the menu shown in the example run below to read, validate and return the user's choice. Depending on this choice, perform the operation mentioned in the menu item selected, write each menu item as a separate function.

numbers.txt

```
15 24  65  2  33  78  5  61  4  42  23  1  12  18  32  68  123  111  75
```

Example Run:

```
DISPLAY
1. All numbers
2. Even numbers
3. Subscripts of odd numbers
4. The numbers with even subscripts
5. Minimum number
6. Subscript of maximum number
7. Exit
Enter your choice: 1

All numbers
************
15 24 65 2 33 78 5 61 4 42 23 1 12 18 32 68 123 111 75

DISPLAY
1. All numbers
2. Even numbers
3. Subscripts of odd numbers
4. The numbers with even subscripts
5. Minimum number
6. Subscript of maximum number
7. Exit
Enter your choice: 2
```

```
Even numbers
************
24 2 78 4 42 12 18 32 68

DISPLAY
1. All numbers
2. Even numbers
3. Subscripts of odd numbers
4. The numbers with even subscripts
5. Minimum number
6. Subscript of maximum number
7. Exit
Enter your choice: 3

Subscripts of odd numbers
*************************
0 2 4 6 7 10 11 16 17 18

DISPLAY
1. All numbers
2. Even numbers
3. Subscripts of odd numbers
4. The numbers with even subscripts
5. Minimum number
6. Subscript of maximum number
7. Exit
Enter your choice: 4

The numbers with even subscripts
*********************************
15 65 33 5 4 23 12 32 123 75

DISPLAY
1. All numbers
2. Even numbers
3. Subscripts of odd numbers
4. The numbers with even subscripts
5. Minimum number
6. Subscript of maximum number
7. Exit
Enter your choice: 5

Minimum number
*********************************
1

DISPLAY
1. All numbers
2. Even numbers
3. Subscripts of odd numbers
4. The numbers with even subscripts
5. Minimum number
6. Subscript of maximum number
7. Exit
Enter your choice: 6

Minimum number
*********************************
16

DISPLAY
1. All numbers
2. Even numbers
3. Subscripts of odd numbers
4. The numbers with even subscripts
5. Minimum number
6. Subscript of maximum number
7. Exit
Enter your choice: 7
```