Department of Information Systems and Technologies

CTIS151 – Introduction to Programming

FALL 2024 - 2025

Lab Guide #13 - Week 10 – 1

OBJECTIVES :
- File Operations and One-Dimensional Arrays

Instructors : Serpil TIN
Assistants : Berk ÖNDER, Efe Mert ŞAHİNKOÇ, Hatice Zehra YILMAZ

Q1.

a) Write a C program in which you declare an integer data type array with a constant size of 10, and initialize it by using the values given to display the array contents on the screen as shown in the example run.

```
int arr[10] = { 78, 43, 1, 12, 90, 34, 55, 65, 75, 99 };
```

Project Name: LG13_Q1a
File Name: Q1a.cpp

Example Run:
```
The array contents are: 78 43 1 12 90 34 55 65 75 99
```

b) Modify your code from part (a) in such a way that the program no longer initializes the array with the values given to you but with those given by the user. Then, display the array contents on the screen.

Example Run:
```
Enter 10 numbers: 65 756 86 96 16 26 36 56 77 88
The array contents are: 65 756 86 96 16 26 36 56 77 88
```

Project Name: LG13_Q1b
File Name: Q1b.cpp

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

values.txt

```
57 21 6 13 145 211 2 65 75 85
```

Project Name: LG13_Q1c
File Name: Q1c.cpp

Example Run:
```
Reading contents of the file, the 10 numbers are: 57 21 6 13 145 211 2 65 75 85
The array contents are: 57 21 6 13 145 211 2 65 75 85
```

Q2. 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

```
In the vast tapestry of existence gratitude is the thread that weaves together moments of joy
resilience and peace illuminating paths that lead us back to what truly matters
```

reverseA.txt:

```
srettam ylurt tahw ot kcab su dael taht shtap gnitanimulli ecaep dna ecneiliser yoj fo stnemom
rehtegot sevaew taht daerht eht si edutitarg ecnetsixe fo yrtsepat tsav eht nI
```

Project Name: LG13_Q2a
File Name: Q2a.cpp

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

reverseB.txt:

```
nI eht tsav yrtsepat fo ecnetsixe edutitarg si eht daerht taht sevaew rehtegot stnemom fo yoj
ecneiliser dna ecaep gnitanimulli shtap taht dael su kcab ot tahw ylurt srettam
```

Project Name: LG13_Q2b
File Name: Q2b.cpp

Q3. Write a C program that generates a random array of integers within the range of 13 to 100, with a size of 15. It prints the original array, adds 20 to each element, and then prints the ASCII equivalent characters of the modified values to an output file named "output.txt." as shown below.

Project Name: LG13_Q3
File Name: Q3.cpp

Example Run:

```
Randomly Generated Array:
81 14 52 57 94 62 76 72 84 84 73 40 71 39 91
```

Output.txt

```
Adding 20 to each then ASCII Equivalent
101 e
34 "
72 H
77 M
114 r
82 R
96 `
92 \
104 h
104 h
93 ]
60 <
91 [
59 ;
111 o
```

Q4. Write a C program that declares an integer array with a **maximum** size of **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 an even number of elements!!!

Write the following function;

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

Project Name: LG13_Q4
File Name: Q4.cpp

Example Run:

```
Enter the number of elements: -5
The number of elements should be EVEN and POSITIVE.

Enter the number of elements: 9
The number of elements should be EVEN and POSITIVE.

Enter the number of elements: 5
The number of elements should be EVEN and POSITIVE.

Enter the number of elements: 8

Enter array elements:
Enter element on index [0]:12
Enter element on index [1]:20
Enter element on index [2]:30
Enter element on index [3]:45
Enter element on index [4]:78
Enter element on index [5]:34
Enter element on index [6]:12
Enter element on index [7]:22

Array elements after swapping adjacent elements:
20  12 45 30 34 78 22 12
```

Additional Questions

AQ1.

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 the file stream and the array as parameters to read the data from the text file into the array. Then, write a function named **menu()** that displays the menu as 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, and 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:

Project Name: LG13_AQ1
File Name: AQ1.cpp

```
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
```

AQ2.

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, 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 a different number of letters, therefore, to determine the word's ending, players must type a dot character '.' right after the word.

Project Name: LG13_AQ2
File Name: AQ2.cpp

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.
```