Department of Information Systems and Technologies

CTIS151 – Introduction to Programming

Fall 2024 – 2025

Lab Guide #18 – Week 13 - 2

OBJECTIVES : Exercises with One-dimensional and Two-dimensional Arrays

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

Q1. Write a C program that declares a 9 x 9 two-dimensional matrix, generates a random number between 1 and 9 as a dimension, and validates whether the number is odd. Then, it fills the array according to the dimension and displays the content of the matrix as in the example run.

Write the following functions;

- **`fillPattern`** that gets a two-dimensional array and a dimension as parameters and fills the array according to the given dimension.
- **`display`** that gets a two-dimensional array as a parameter and displays the content of it.

Example Run #1:

```
Generated random dimension : 6
Generated random dimension : 3

      MATRIX
 0 1 0 0 0 0 0 0 0
 1 1 1 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
```

Example Run #2:

```
Generated random dimension : 6
Generated random dimension : 5

      MATRIX
 0 0 1 0 0 0 0 0 0
 0 1 1 1 0 0 0 0 0
 1 1 1 1 1 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0
```

Project Name: LG18_Q1
File Name: Q1.cpp

Q2. Write a C program that takes two words, checks if they are equal, and displays their equality status. Also, if the words are not equal, the program displays the number of characters from the beginning that differ using the functions below.
Note: Use newline **('\n')** characters to determine the end of a word.

Write the following functions;

- **`displayWord`** that displays a given word.
- **`isEqual`** that takes two words along with their sizes and checks for if the words are the same or not. The function returns the position of the first unmatched character.

Example Run #1:

```
Enter first word: sacrifice
Enter second word: ice

sacrifice and ice are NOT the same words.
First different character in between was after the character #0.
```

Example Run #2:

```
Enter first word: outliers
Enter second word: outliers

outliers and outliers are the same words.
```

Example Run #3:

```
Enter first word: rich
Enter second word: risk

rich and risk are NOT the same words.
First different character in between was after the character #2.
```

Project Name:
LG18_Q2

Q3. Write a C program that finds the abbreviation of each line read from the text file named **text.txt** using the function below. (Assume that there is a newline character ('\n') at the end of each line) The output of the function should be written to a text file called "**abbreviation.txt**".

Write the following function;

- that takes a character array and an actual number of characters in the array as input parameters, finds and returns the abbreviated charater array and its size.

Project Name: LG18_Q3
File Name: Q3.cpp

text.txt

```
Don't Bother Me I'm Busy
Early Morning Business Meeting
Good Game
In Real Life
Laughing Out Loud
I Miss You
```

abbreviation.txt

```
DBMIB
EMBM
GG
IRL
LOL
IMY
```

Additional Questions

AQ1. There was an election in two cities Ankara and İstanbul. 5 political parties joined the election. The information on the election is given in the file named "**votes.txt"**. Each row contains the party number, number of votes, and the city code (A-Ankara, I-İstanbul).

Write a modular C program that reads the number of votes received by 5 political parties in the election from the text file into a two-dimensional integer array and performs the following operations:

- Prints the total number of votes for each city.
- Prints the number of votes for each party and the percentage of the total votes for each city they received.

Write the following functions;

- **`findTotVote`** that gets a two-dimensional array and a column number as parameters, finds and returns the total votes in the given city according to the column number.
- **`findPercentage`** that gets the number of votes and the total votes in a certain city as parameters, finds and returns the percentage of the total votes.

Project Name: LG18_AQ1
File Name: AQ1.cpp

Example Run :

```
City code       ANKARA        ISTANBUL
*********     **********     **********
  Total          2688           1640
*********     **********     **********
    1         854  31.8%     365  22.3%
    2         484  18.0%     281  17.1%
    3         790  29.4%     340  20.7%
    4         491  18.3%     147   9.0%
    5          69   2.6%     507  30.9%
```

votes.txt

```
1 156 A
2 245 I
3 550 A
3 340 I
2 142 A
1 365 I
4 241 A
5 365 I
5 69 A
4 147 I
2 36 I
3 240 A
4 250 A
5 142 I
1 698 A
2 342 A
```

AQ2. In role-playing games (RPGs), game characters have attributes such as strength, dexterity, and intelligence, represented by integer values. Below is a table with sample character data. The character with ID 111 has 10 strength points, 5 dexterity points, and 3 intelligence points.

ID	STRENGTH	DEXTERITY	INTELLIGENCE
111	10	5	3
222	9	8	7
333	1	11	19
444	8	2	15
555	9	9	2
666	3	7	1

Write a modular C program that reads character data from the text file named **"info.txt"** for 6 characters into an integer matrix. The data includes the character's **ID, strength, dexterity,** and **intelligence points** in the given order. The program then displays a menu, inputs a choice and calls the corresponding functions until the user selects choice 7.

Write the following functions;

- **`menu()`**
  1) Display character info
  2) Update strength of all characters
  3) Find id of character with lowest dexterity
  4) Find id of character with highest intelligence
  5) Display all characters
  6) Calculate armor ratings
  7) Exit
- **`displayCharInfo()`** that takes matrix and id as parameters Searches the id in the matrix (only on column 0) and displays the row that the id belongs to.
- **`updateStrength()`** that takes a matrix as a parameter and asks the user to input a number. Then, increases the strength of each character by that number.
- **`findLowestDex()`** that takes a matrix as a parameter, finds and returns the index of the character with the lowest dexterity points.
- **`findHighestInt()`** that takes a matrix as a parameter, finds and returns the index of the character with the highest intelligence points.
- **`calculateArmorRating()`** that takes a matrix, armor rating multipliers as an array, and a one-dimensional array as an output parameter, calculates the armor rating for each character and returns them in an array. (Armor rating multipliers: Strength 3, Dexterity 2, Intelligence 1)
- **`displayAll()`** that displays all characters' info on the screen as a table.

`info.txt`

```
111 10 5 3
222 9 8 7
333 1 11 19
444 8 2 15
555 9 9 2
666 3 7 1
```

Project Name: LG18_AQ2
File Name: AQ2.cpp

Example Run:

```
Menu:
1) Display character info
2) Update strength of all characters
3) Find id of character with lowest dexterity
4) Find id of character with highest intelligence
5) Display all characters
6) Calculate armor ratings
7) Exit
Enter your choice: 1
Enter character ID to display info: 111
ID: 111, Strength: 10, Dexterity: 5, Intelligence: 3

Menu:
1) Display character info
2) Update strength of all characters
3) Find id of character with lowest dexterity
4) Find id of character with highest intelligence
5) Display all characters
6) Calculate armor ratings
7) Exit
Enter your choice: 3
Character with lowest dexterity has ID: 444

Menu:
1) Display character info
2) Update strength of all characters
3) Find id of character with lowest dexterity
4) Find id of character with highest intelligence
5) Display all characters
6) Calculate armor ratings
7) Exit
Enter your choice: 4
Character with highest intelligence has ID: 333

Menu:
1) Display character info
2) Update strength of all characters
3) Find id of character with lowest dexterity
4) Find id of character with highest intelligence
5) Display all characters
6) Calculate armor ratings
7) Exit
Enter your choice: 6
Armor Ratings:
ID 111: 43
ID 222: 50
ID 333: 44
ID 444: 43
ID 555: 47
ID 666: 24

Menu:
1) Display character info
2) Update strength of all characters
3) Find id of character with lowest dexterity
4) Find id of character with highest intelligence
5) Display all characters
6) Calculate armor ratings
7) Exit
Enter your choice: 5
ID      Strength        Dexterity       Intelligence
111     10              5               3
222     9               8               7
333     1               11              19
444     8               2               15
555     9               9               2
666     3               7               1
```

```
Menu:
1) Display character info
2) Update strength of all characters
3) Find id of character with lowest dexterity
4) Find id of character with highest intelligence
5) Display all characters
6) Calculate armor ratings
7) Exit
Enter your choice: 2
Enter the number to increase strength by: 2
Strength updated.

Menu:
1) Display character info
2) Update strength of all characters
3) Find id of character with lowest dexterity
4) Find id of character with highest intelligence
5) Display all characters
6) Calculate armor ratings
7) Exit
Enter your choice: 5
ID      Strength        Dexterity       Intelligence
111     12              5               3
222     11              8               7
333     3               11              19
444     10              2               15
555     11              9               2
666     5               7               1

Menu:
1) Display character info
2) Update strength of all characters
3) Find id of character with lowest dexterity
4) Find id of character with highest intelligence
5) Display all characters
6) Calculate armor ratings
7) Exit
Enter your choice: 7
Exiting program.
```