LAB 07: Arrays

1. Scope of Knowledge:

- Understand the concept of arrays
- Declare and initialize arrays in C (1-dimensional array, 2-dimensional array)
- Working with arrays

2. Marterials/Softwares/Tools:

- Visual Studio Code
- Draw IO (online) or Microsoft Word

3. Coding Convention:

- All identifiers must be in English and lower case
- Follow the valid identifers naming rules in C
- Tab is 4 characters
- Curly braces must be aligned
- Statements in curly brackets must be indented by 1 tab

4. Exercise:

Note: Students must draw a flowchart describing the algorithm before programming for each problem.

Exercise 1:

Draw a flowchart and write a program to declare an integer array with 10 elements, then enter data for array's elements and print all the elements in even positions (index = 0, 2, 4, ...).

Exercise 2:

Draw a flowchart and write a program to declare an array of integers (int) with 10 elements and an array of real numbers (float) with 5 elements. Input the array of real

numbers then assign the elements of this array to the odd positions of the integer array (remaining elements are zero). Finally display these two arrays on the screen.

Exercise 3:

Draw a flowchart and write a program to declare an integer array of 10 elements, enter data for this array and then print the maximum, minimum and number of their occurrences in the array.

For example:

Integer Array = {12, 8, 23, 8, 25, 8, 90, 21, 11, 90}

The maximum value is 90, the number of occurrences is 2.

The maximum value is 8, the number of occurrences 3.

Exercise 4:

Draw a flowchart and write a program:

Declare an integer array with 5 elements, ask the user to input from the keyboard a prime number, if not a prime number requires re-entering.

Exercise 5:

Draw a flowchart and write a program to declare an integer array of 10 elements, take a random number in the range [0-100] and assign it to the elements of this array. Then enter a number in the range [0-100]. Display a list of random numbers, check if the number is in the random number that has just been generated or not? the position of the first occurrence of the entered number and the number of occurrences of this number in that sequence.

Hint:

Use the srand() and rand() functions which are declared in 2 libraries <time.h> and <stdlib.h>:

```
srand(time(NULL) );

// random will receive a random value from 0 to 9

int random = rand() % 10;
```

Exercise 6:

Write a program that declares an integer array of 20 elements to store the first 20 elements of the Fibonacci sequence.

Exercise 7*:

With the requirements to manage the array of prime numbers as in post number 5 above, however, the number of elements increases to 20 and the random element of the array must be a prime number in the range [0-200].

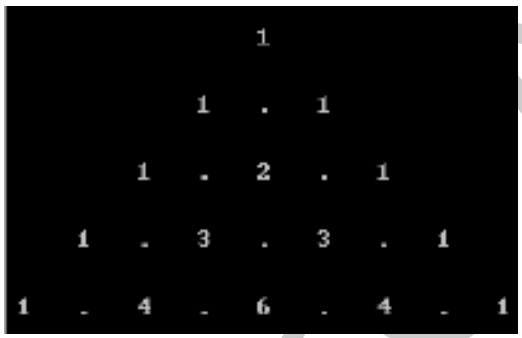
Exercise 8*:

Write a program to manage a character array of 10 elements. Enter from the keyboard the characters of the alphabet from a-z, convert the elements in the array to uppercase (A-Z) and print the result to the screen.

Exercise 9*:

Use a 2-dimensional array of size [10 x 10] to display Pascal's triangle with height 5: In Pascal's triangle, the two sided elements are all 1, from the 3rd line onwards the 2nd element of this line is equal to the sum of the first and 2nd elements of the previous row.

General: a[i][j] = a[i-1][j-1] + a[i-1][j] where i>1, j>0.



Exercise 10*:

To store the scores of 6 subjects in semester 1 for 3 students, a 2-dimensional array of 3 rows and 6 columns is used. Write a program to input the scores for those 3 students, displaying the grade and average score of all above students in the table.

Student	CF	BPL	OOP	NPF	RDBMS	AVG	GRADE
Student 1	20.5	16.0	12.0	15.0	19.0	16.5	C
Student 2	10.0	10.5	11.0	11.5	12.0	11.0	F
Student 3	18.0	21.0	19.5	24.0	22.0	20.9	B

Hint:

Mark is from 0 to 25. You must convert mark to percentage to calculate the grade.

Grade types:

- average mark < 50% → grade F
- 50 <= average mark < 65 → grade D
- 65 <= average mark < 80 → grade C
- 80 <= average mark < 90 → grade B
- 90 <= average mark <= 100 → grade A