TOPICS: Arrays-II (Multi-dimentional) **Allowed Time:** 60 Minutes
Instructions: **Total Marks:** 40

1. Gossips are not allowed.
2. Teacher assistants are for your help, so be nice with them. Respect them as they are teaching you. Raise your hands if you have some problem and need help from TA. Avoid calling them by raising your voice and disturbing the environment of Lab.
3. TA may deduct your marks for any kind of ill-discipline or misconduct from your side.
4. Evaluation will be considered final and you cannot debate for the marks. So, focus on performing the tasks when the time is given to you.

Task 01: (10 Marks, 10 min)

Create a C program that:

- Defines a character **array of size 10**
- Sets values of array by taking a string literal as input from user
- Finds the frequency of each letter and display it on console
- Finds the most frequent letter and displays it on console
- Convert the string to uppercase and print on console
- Convert the string to lowercase and print on console

Task 02: (15 Marks, 20 min)

Create a C program that:

- Creates a 2-D matrix of size 3x3.
- Fills the matrix with user given values.

Determine then following:

- Whether the matrix is **diagonal matrix** or not.
- Whether the matrix **is identity matrix** or not.

A diagonal matrix is a matrix in which the entries outside the main diagonal are all zero. The diagonal entries themselves may or may not be zero.

The identity matrix or unit matrix of size n is the n × n square matrix with ones on the main diagonal and zeros elsewhere.

Sample Output:

```
Microsoft Visual Studio Debug Console
Enter Row 1 elements : 1 0 0
Enter Row 2 elements : 0 2 0
Enter Row 3 elements : 0 0 3

1  0  0
0  2  0
0  0  3
Its a Diagonal matrix !
```

```
Microsoft Visual Studio Debug Console
Row 1 elements : 1 0 0
Row 2 elements : 0 1 0
Row 3 elements : 0 0 1

1  0  0
0  1  0
0  0  1
Its a identity matrix !
```

Task 03: (15 Marks, 20 min)

Write C function that displays a **N × N matrix (array)** on screen in a neat and readable way.
After that, write another function that swaps the contents of the main diagonal (which runs from top-left to bottom-right corner) of the matrix with the contents of the antidiagonal (which runs from top-right to bottom-left corner).
Finally, your program should once again display the (now modified) N × N matrix (array) on screen.
Important Note: You MUST implement the logic of your program using at least 2 different functions (apart from the main function).

For example, if N is 7 then your program should display the following matrices on screen:

```
Initial matrix (BEFORE swapping diagonals):

 1  8 15 22 29 36 43
 2  9 16 23 30 37 44
 3 10 17 24 31 38 45
 4 11 18 25 32 39 46
 5 12 19 26 33 40 47
 6 13 20 27 34 41 48
 7 14 21 28 35 42 49


Final matrix (AFTER swapping diagonals):

43  8 15 22 29 36  1
 2 37 16 23 30  9 44
 3 10 31 24 17 38 45
 4 11 18 25 32 39 46
 5 12 33 26 19 40 47
 6 41 20 27 34 13 48
49 14 21 28 35 42  7
```

Note that the elements on the main diagonal have been put in BLUE color (1,9,17,33,41,49 after swapping 43,37,31,19,13,7), elements
on the antidiagonal have been put in GREEN color, (43,37,31,19,13,7 after swapping 1 ,9,17,33,41,49), and the common element

between the two diagonals (i.e. the center-most element) has been put in RED color (25).
These colors have been used just for your understandability, and the output produced by your program will (obviously) not be colored.