BASIS AND PRACTICE IN C PROGRAMMING
SPRING SEMESTER 2022

INSTRUCTOR: Prof. TAMER ABUHMED
COLLEGE OF SOFTWARE

Assignment 3

This assignment consists of 3 tasks. Guidelines for submission format are given at the end of the assignment file.

Note: Texts in green color should be inserted by the keyboard. Follow the submission, example output formats, and task description in order to get a full score.

Task 1. Write a program that adds two matrices. Calculate the matrix addition in a separate **function**.

Example input from keyboard:

```
Enter number of rows:  3
Enter number of columns:  3
Enter the first matrix elements:
1  1  1
2  2  2
3  3  3
Enter the second matrix elements:
1  1  1
2  2  2
3  3  3
Addition of the matrices:
2  2  2
4  4  4
6  6  6
```

Task 2. Write a program that counts the occurrence of lowercase, uppercase, special characters, and numeric values from a given string. Write the logic in a **separate** function.

Example input from keyboard 1:

```
Enter the string: #helLo23wOr@17
```

Example output 1:

```
Uppercase letters: 2
Lowercase letters: 6
Numbers: 4
Special characters: 2
```

Task 3. Write a program that sorts the given array in ascending order so that the elements of the array will be arranged from the smallest to the largest. Write the logic in a **separate** function.

Example input from keyboard:

```
Enter the elements of the array: 5 8 6 1
```

Example output:

```
The result: 1 5 6 8
```

Submission format: Submit 3 separate C files. C files must include the implementation code of each task and comments for **important** lines of code to explain the purpose. All the files should be submitted as a **zip** file.

Name of zip file: {student ID}_{Student name}_assignment3.zip
Example: 2020712837_Frank_Thomas_assignment3.zip

Example for source code submission format in C files:

```
/*
 * Function: square_the_biggest
 * ----------------------------
 *   Returns the square of the largest of its two input values
 *
 *   n1: one real value
 *   n2: the other real value
 *
 *   returns: the square of the larger of n1 and n2
 */

double square_the_biggest (float n1, float n2) {
    if(n1 > n2) {          // check if the first number is bigger
       return n1*n1;   // if the first number is bigger, then return square of n1
    }
    return n2*n2;       // otherwise return square of n2
}
```