Lab 1: Review

Write programs/functions to fulfill the following requirements: *(Note: the given function prototype is for reference only)*

1 Function

1. Solving the linear equation: $ax + b = 0$.

   ```
   void LinearEquation(int a, int b)
   ```

2. Solving the quadratic equation: $ax^2 + bx + c = 0$.

   ```
   void QuadraticEquation(int a, int b, int c)
   ```

3. Determine if an given integer is a prime.

   ```
   bool isPrime(int a)
   ```

4. Count the number of prime between 2 input integer value a and b (a < b).

   ```
   int countPrime(int a, int b)
   ```

5. Calculate the total value of all digits of a given integer.

   ```
   int sumDigits(int n)
   ```

6. Count the number of integer that is smaller than an input value N and divisible by k.

   ```
   int countInteger(int N, int k)
   ```

7. Given a, b, and c are 3 real numbers. Determine if a, b and c form a triangle. Which type of triangle is that? (normal triangle, right triangle, isosceles triangle, right isosceles triangle, equilateral triangle)

   ```
   void Triangle(float a, float b, float c)
   ```

2 Array & String

2.1 Array

- Input an array with size n.

  ```
  void inputArray(float A[], int n)
  ```

- Output a given array with size n.

  ```
  void printArray(float A[], int n)
  ```

- Count the number of prime in a given array.

```
int countArrayPrime(int A[], int n)
```

- Calculate the summarize of all elements from a given array.

```
float sumArray(float A[], int n)
```

- Determine if a given array is increasing / decreasing.

```
bool isIncreasing(float A[], int n)
bool isDecreasing(float A[], int n)
```

2.2 2D Array

- Input an 2D array with size m `x` n.

```
void input2DArray(float A[][], int m, int n)
```

- Output a given 2D array with size m `x` n.

```
void print2DArray(float A[][], int m, int n)
```

- Rotate a given 2D array 90^o clockwise.

```
void rotate2DArray(float A[][], int m, int n)
```

- Calculate the summarize of 2 given 2D arrays.

```
void sum2DArray(float A[][], float B[][], float result[][], int m, int n)
```

- Calculate the multiplication of 2 given 2D array.

```
void multiple2DArray(float A[][], float B[][], float result[][], int m, int n, int p)
```

- Determine if a given array is a diagonal matrix / upper triangle matrix / lower triangle matrix.

```
bool isDiagonalMatrix(float A, int m, int n)
bool isUpperTriangleMatrix(float A, int m, int n)
bool isLowerTriangleMatrix(float A, int m, int n)
```

2.3 String

- Input a string and print it out on the screen.

  `void inputString(char C[100])`

- Determine if a given string is palindrome.

  `void printString(char C[100])`

- Count the number of capital characters from a given string.

  `int countCapital(char C[100])`

- Count the number of appearances of an input character from a given string.

  `int countAppearance(char C[100]), char c`

- Count the number of words from a given string.

  `int countWord(char C[100])`

3 Structure

Declare the following structures and complete its function

1. A structure represent the point of time of a day (from 00:00:00 to 23:59:59)
   - Input a point of time, verify the input.
   - Given a point of time, calculate the number of minutes and second of the day has passed
   - Given 2 point of time, determine which one is earlier.
   - Given a point of time, calculate the next point of time after adding x given minutes.
2. A structure represent a fraction (numerator and denominator must be integer)
   - Input a fraction, verify the input.
   - Simplify a given fraction.
   - Calculate the total fraction of 2 given fractions.
   - Compare 2 given fractions.
3. A structure represent a point in an 2D coordinate.
   - Determine which quadrant is a given point positioned. (up-left, up-right, down-left, down-right)
   - Calculate the distance between 2 given points.
   - Determine the midpoint of a line created by 2 given points.
   - Determine if 3 given points are collineared.

4 File

1. Given the *"input1.txt"* file with the following structure:

   - 1^{st} line: integer n
   - Next n lines: each contains n real single equations. Example:

     3
     - $1.1 + 3.5$
     - $4.4 * 8$
     - $1.2/2.0$

   Generate the *"output1.txt"* file with the result of equations from the *"input1.txt"*, each equation's result on a single line

2. Given the *"input2.txt"* file. Let the user input a word and count the number of appearances of the word from the given file.