Subject: PRF192 - Programming Fundamental with C
Workshop 5

Objectives:

In this workshop, you will:

- To understand how to manipulate files in C programming.
- To learn how to read numerical data from a text file into an array.
- To apply concepts of arrays, loops, and file I/O operations in solving real-world problems.
- To enhance problem-solving skills by processing and analyzing data stored in external files.

Problem Overview:

Design a program that reads numerical data from a text file into an array. The program will perform the following tasks:

1. Read data from a text file into an array.
2. Display the data stored in the array.
3. Calculate and display the sum, average, maximum, and minimum values of the array elements.
4. Write the results (sum, average, max, and min) to another text file.

Situation Description:

In real-world scenarios, numerical data is often stored in external files, such as logs, datasets, or reports. For example:

- A financial system may store daily sales in a file that needs to be analyzed for trends.
- A weather monitoring system may save temperature readings in a file for later statistical analysis.

This program simulates such a scenario by working with a text file containing numerical data. The program processes this data to generate meaningful insights.

Syntax Use in the Problem:

1. **File Handling**:

- Use `fopen` to open a file for reading and writing.
- Use `fscanf` to read numbers from the file.
- Use `fprintf` to write results to another file.

2. **Loops**:

   - A `while` loop reads the file data into the array until the end of the file.
   - A `for` loop processes the array to calculate the required statistics.

3. **Conditional Statements**:

   - Use `if` conditions to find the maximum and minimum values in the array.

Specific Requirements:

1. **Input:**

The input text file (**data.txt**) contains one number per line.

2. **Output:**

   - Display the numbers stored in the array.
   - Print the sum, average, maximum, and minimum values to the console.
   - Write the results to an output file (**results.txt**).

3. **File Handling:**

   - Validate file opening; if a file cannot be opened, display an error message and terminate the program.

Sample Input File (“data.txt”):

```
25
30
45
10
60
5
```

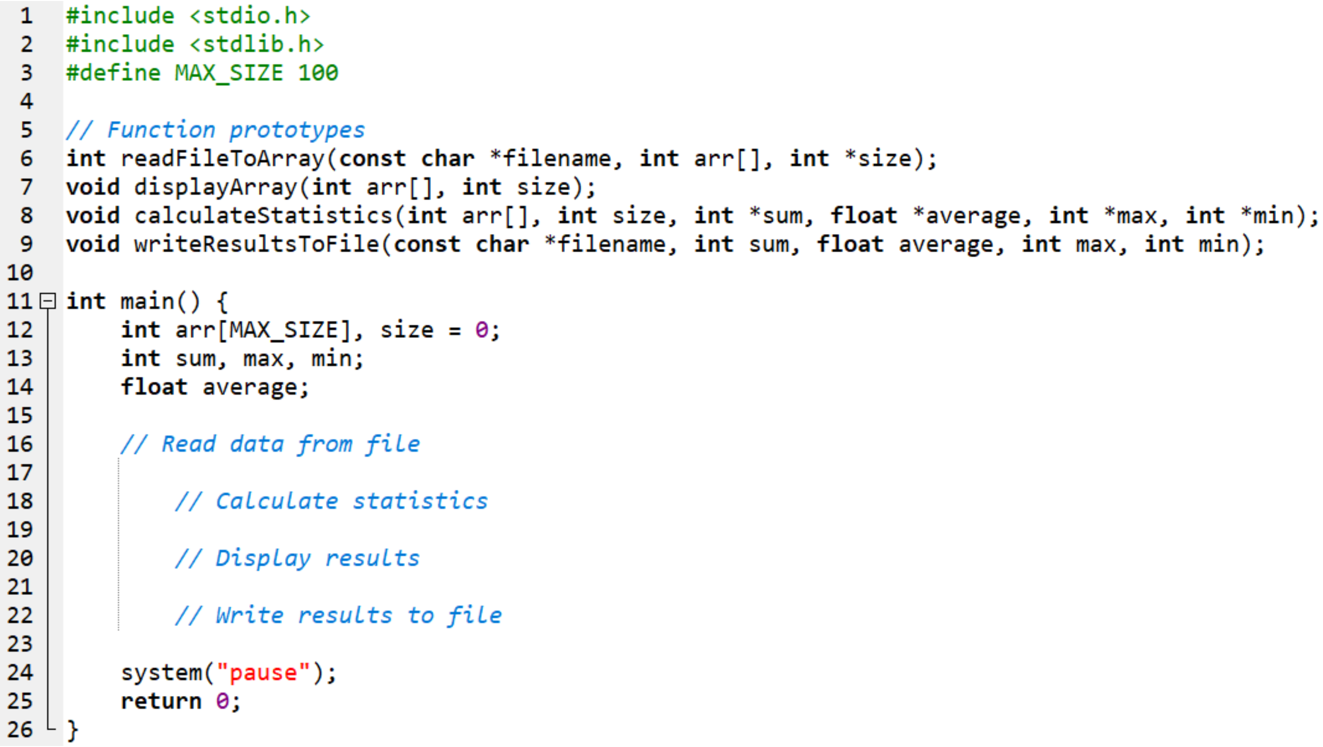
Expected Console Output:

```
Data in the array:
25 30 45 10 60 5
Statistics:
Sum: 175
Average: 29.17
Maximum: 60
Minimum: 5
Results written to results.txt
```

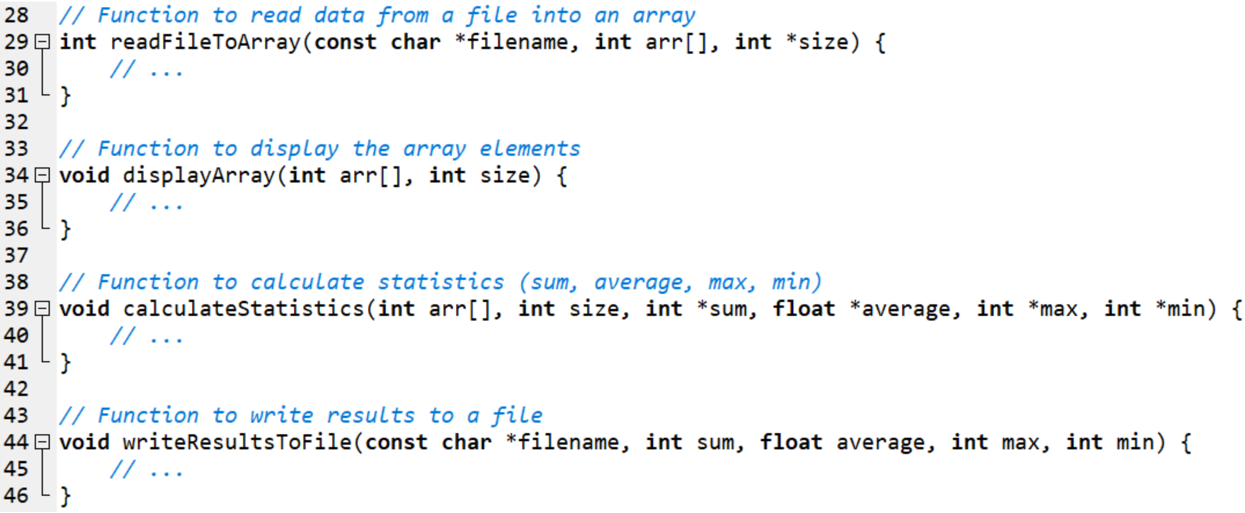
Sample Output File (“results.txt”):

```
Sum: 175
Average: 29.17
Maximum: 60
Minimum: 5
```

Hint: Code Design

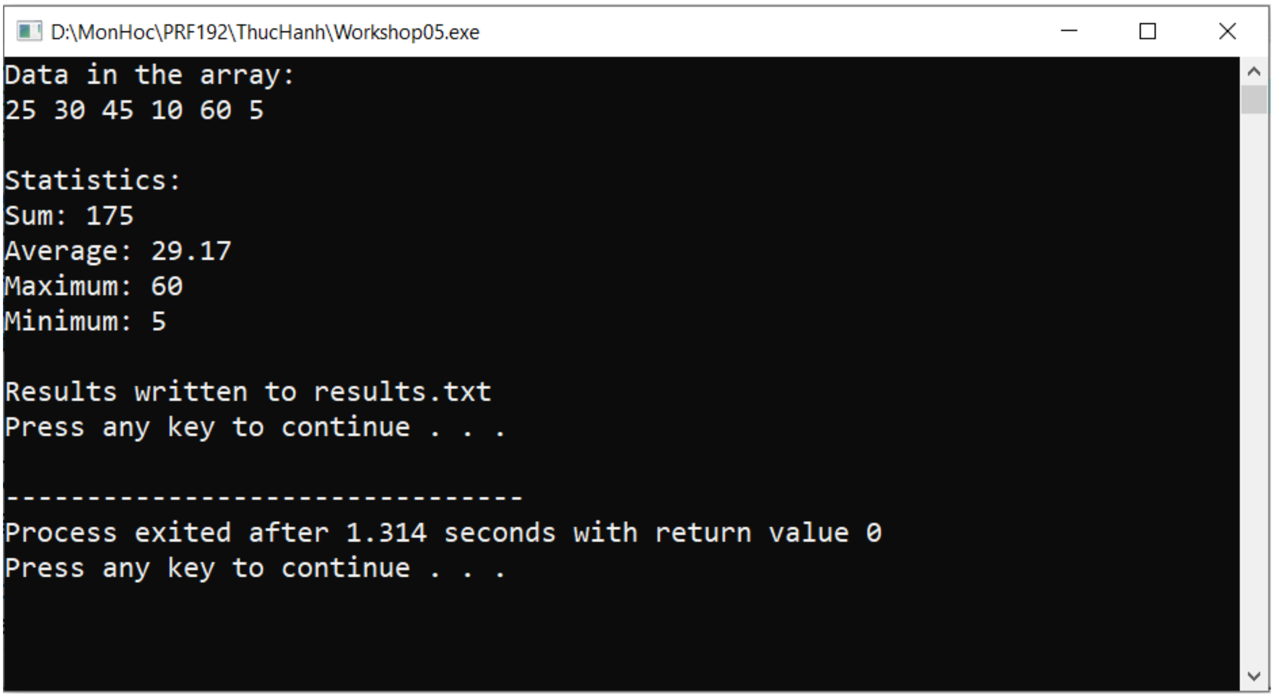


```c
#include <stdio.h>
#include <stdlib.h>
#define MAX_SIZE 100

// Function prototypes
int readFileToArray(const char *filename, int arr[], int *size);
void displayArray(int arr[], int size);
void calculateStatistics(int arr[], int size, int *sum, float *average, int *max, int *min);
void writeResultsToFile(const char *filename, int sum, float average, int max, int min);

int main() {
    int arr[MAX_SIZE], size = 0;
    int sum, max, min;
    float average;

    // Read data from file

        // Calculate statistics

        // Display results

        // Write results to file

    system("pause");
    return 0;
}
```



```
// Function to read data from a file into an array
int readFileToArray(const char *filename, int arr[], int *size) {
    // ...
}

// Function to display the array elements
void displayArray(int arr[], int size) {
    // ...
}

// Function to calculate statistics (sum, average, max, min)
void calculateStatistics(int arr[], int size, int *sum, float *average, int *max, int *min) {
    // ...
}

// Function to write results to a file
void writeResultsToFile(const char *filename, int sum, float average, int max, int min) {
    // ...
}
```

Output Sample:



```
Data in the array:
25 30 45 10 60 5

Statistics:
Sum: 175
Average: 29.17
Maximum: 60
Minimum: 5

Results written to results.txt
Press any key to continue . . .

--------------------------------
Process exited after 1.314 seconds with return value 0
Press any key to continue . . .
```