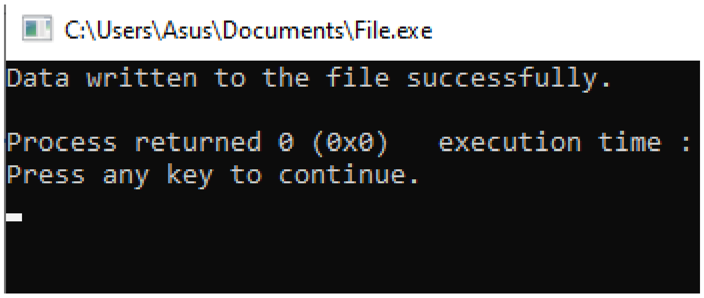
1. Write a program in C to create and store information in a text file.
   Input:

```
#include <stdio.h>

int main() {
   // Declare a FILE pointer
   FILE *file;

   // Open the file in write mode ("w")
   file = fopen("example.txt", "w");

   // Check if the file was successfully opened
   if (file == NULL) {
      printf("Error opening the file.\n");
      return 1; // Return an error code
   }

   // Write information to the file
   fprintf(file, "Name: John Doe\n");
   fprintf(file, "Age: 25\n");
   fprintf(file, "Occupation: Programmer\n");

   // Close the file
   fclose(file);

   printf("Data written to the file successfully.\n");

   return 0; // Return 0 to indicate successful execution
}
```

Output:



```
C:\Users\Asus\Documents\File.exe
Data written to the file successfully.

Process returned 0 (0x0)   execution time :
Press any key to continue.
```

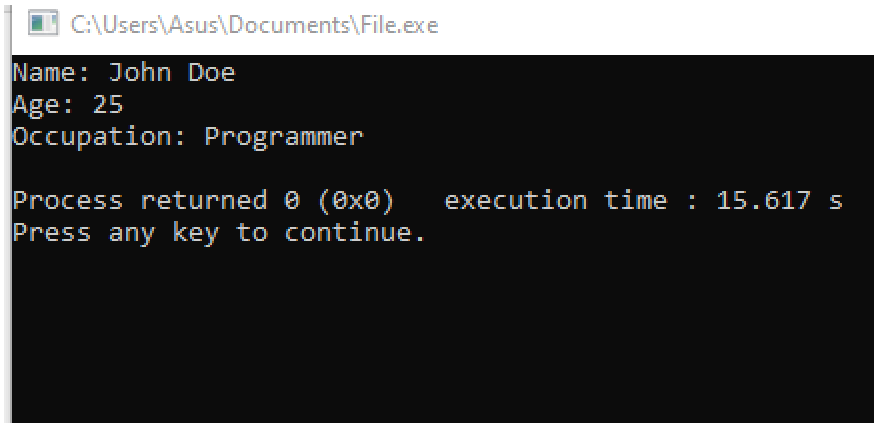
2. Write a program in C to read an existing file and print the text in the console.

Input:

```
#include <stdio.h>

int main() {
   // Declare a FILE pointer
   FILE *file;

   // Open the existing file in read mode ("r")
   file = fopen("example.txt", "r");

   // Check if the file was successfully opened
   if (file == NULL) {
      printf("Error opening the file.\n");
      return 1; // Return an error code
   }

   // Read and print each line of the file
   char line[100]; // Assuming each line in the file is less than 100
characters
   while (fgets(line, sizeof(line), file) != NULL) {
      printf("%s", line);
```

```
    }

    // Close the file
    fclose(file);

    return 0; // Return 0 to indicate successful execution
}
```

Output:



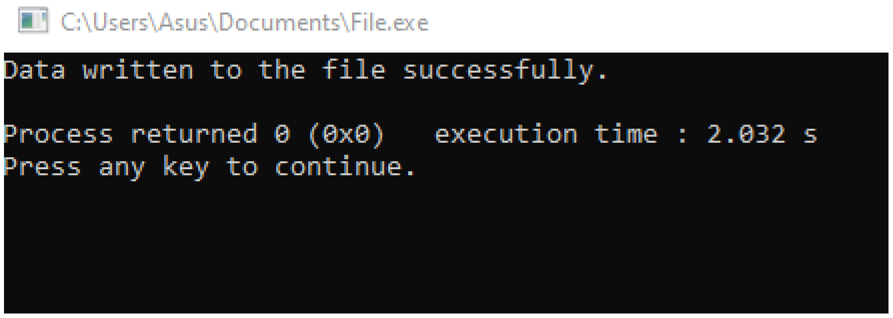
C:\Users\Asus\Documents\File.exe

```
Name: John Doe
Age: 25
Occupation: Programmer

Process returned 0 (0x0)   execution time : 15.617 s
Press any key to continue.
```

3. Write a program in C to write multiple lines to a text file.
   Input:

```
#include <stdio.h>

int main() {
    // Declare a FILE pointer
    FILE *file;

    // Open the file in write mode ("w")
    file = fopen("example_multiline.txt", "w");

    // Check if the file was successfully opened
    if (file == NULL) {
        printf("Error opening the file.\n");
        return 1; // Return an error code
    }

    // Write multiple lines to the file
    fprintf(file, "Line 1: This is the first line.\n");
    fprintf(file, "Line 2: Writing multiple lines to a text file.\n");
    fprintf(file, "Line 3: You can add more lines as needed.\n");
    fprintf(file, "Line 4: Remember to close the file when done.\n");

    // Close the file
    fclose(file);

    printf("Data written to the file successfully.\n");

    return 0; // Return 0 to indicate successful execution
}
```

Output:



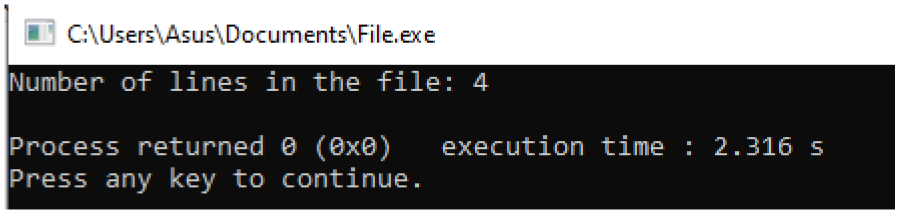
C:\Users\Asus\Documents\File.exe

```
Data written to the file successfully.

Process returned 0 (0x0)   execution time : 2.032 s
Press any key to continue.
```

4. Write a program in C to find the number of lines in a text file.
   Input:

```
#include <stdio.h>

int main() {
    // Declare a FILE pointer
    FILE *file;

    // Open the file in read mode ("r")
    file = fopen("example_multiline.txt", "r");

    // Check if the file was successfully opened
    if (file == NULL) {
        printf("Error opening the file.\n");
        return 1; // Return an error code
    }

    // Count the number of lines in the file
    int lineCount = 0;
    char ch;

    while ((ch = fgetc(file)) != EOF) {
        if (ch == '\n') {
            lineCount++;
        }
    }

    // Close the file
    fclose(file);

    // Print the number of lines
    printf("Number of lines in the file: %d\n", lineCount);

    return 0; // Return 0 to indicate successful execution
}
```

Output:


C:\Users\Asus\Documents\File.exe

```
Number of lines in the file: 4

Process returned 0 (0x0)   execution time : 2.316 s
Press any key to continue.
```

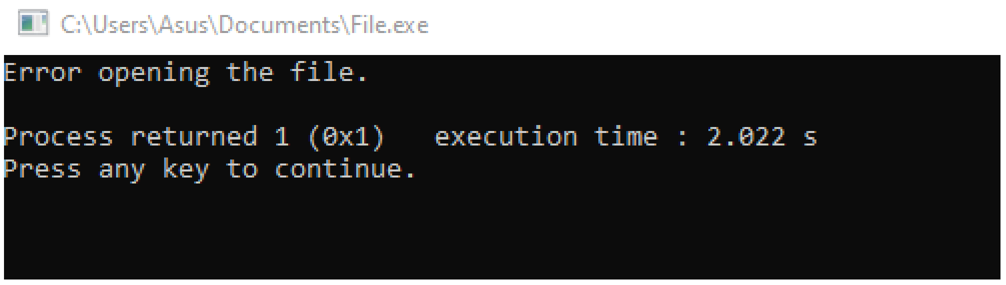
5. A file contains some integer numbers separated by spaces. Write a c program to calculate the total numbers in the files, the sum of those integer numbers, and the average of those numbers.

   Input:

```
#include <stdio.h>

int main() {
   // Declare a FILE pointer
   FILE *file;

   // Open the file in read mode ("r")
   file = fopen("numbers.txt", "r");

   // Check if the file was successfully opened
   if (file == NULL) {
      printf("Error opening the file.\n");
      return 1; // Return an error code
   }

   // Variables to store information
   int number;
   int totalNumbers = 0;
   int sum = 0;

   // Read numbers from the file and calculate the sum
```

```
    while (fscanf(file, "%d", &number) == 1) {
        totalNumbers++;
        sum += number;
    }

    // Close the file
    fclose(file);

    // Calculate the average
    float average = (float)sum / totalNumbers;

    // Print the results
    printf("Total numbers in the file: %d\n", totalNumbers);
    printf("Sum of the numbers: %d\n", sum);
    printf("Average of the numbers: %.2f\n", average);

    return 0; // Return 0 to indicate successful execution
}
```

Output:



C:\Users\Asus\Documents\File.exe

```
Error opening the file.

Process returned 1 (0x1)   execution time : 2.022 s
Press any key to continue.
```