C - File I/O

Last Updated : 23 Sep, 2024

In this article, we will learn how to operate over files using a C program. A single C file can read, write, move, and create files in our computer easily using a few functions and elements included in the C File I/O system. We can easily manipulate data in a file regardless of whether the file is a text file or a binary file using functions like fopen(), fclose(), fprintf(), fscanf(), getc(), putc(), getw(), fseek(), etc.

What are files in C?

A ***file*** is used to store huge data. C provides multiple file management functions like file creation, opening and reading files, Writing to the file, and closing a file. The file is used to store relevant data and file handling in C is used to manipulate the data.

Types of Files in C

There are mainly two types of files that can be handled using File Handling in C as mentioned below:

1. Text Files
2. Binary Files

1. Text Files

These are simple text files that are saved by the (.txt) extension and can be created or modified by any text editor. Text file stores data in the form of ASCII characters and is used to store a stream of characters.

2. Binary Files

stored in binary form i.e, (0's and 1's).

C Files Operations

C Files can perform multiple useful operations that are mentioned below:

1. *Creation of a new file (fopen with attributes as “a” or “a+” or “w” or “w+”).*
2. *Opening an existing file (fopenopen).*
3. *Reading from file (fscanf or fgets).*
4. *Writing to a file with fprintf or fputs.*
5. *Moving file pointer associated with a given file to a specific position. (fseek, rewind).*
6. *Closing a file (fclose).*

File Pointer declaration

For performing operations on the file, a special pointer called File pointer is used that can be declared as:

```
FILE file_ptr;
```

We can open the file as

```
file_ptr = fopen("fileName.txt", "w");
```

File Modes in C

The second parameter i.e, "w" can be changed according to the table below:

Opening Modes	Description
r	Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer that points to the first character in it. If the file cannot be opened fopen( ) then it returns NULL.

Opening Modes	Description
rb	Open for reading in binary mode. If the file does not exist then fopen( ) will return NULL.
w	Searches file. Contents are overwritten if the file exists. A new file is created if the file doesn't exist. Returns NULL, if unable to open the file.
wb	Open for writing in binary mode. Its contents are overwritten if the file exists, else the file will be created.
a	Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer that points to the last character in it. If the file doesn't exist, a new file is created. Returns NULL, if unable to open the file.
ab	Open for append in binary mode. Data is added to the end of the file. A file will be created if it does not exist.
r+	Searches file. It is opened successfully fopen( ) loads it into memory and sets up a pointer that points to the first character in it. If it is unable to open the file it Returns NULL.
rb+	Open for both reading and writing in binary mode. fopen( ) returns NULL if the file does not exist.
w+	Searches file. Its contents are overwritten if the file exists. A new file is created if the file doesn't exist. Returns NULL, if unable to open the file.
wb+	Open for both reading and writing in binary mode. Contents are overwritten if the file exists. It will be created if the file does not exist.

Opening Modes	Description
a+	Searches file. If the file is opened successfully fopen( ) loads it into memory and sets up a pointer that points to the last character in it. If the file doesn't exist, a new file is created. Returns NULL, if unable to open the file.
ab+	Open for both reading and appending in binary mode. A file will be created if the file does not exist.

If you want to handle binary files then access modes like "rb", "wb", "ab", "rb+", r+b", "wb+", "w+b", "ab+", "a+b" will be used mostly.

Opening a file in C

To perform the opening and creation of a file in c we can use the fopen() function which comes under stdio.h header file.

Syntax:

```
p = fopen("fileopen", "mode");
```

Example:

```
p = fopen("Hello.txt", r);
```

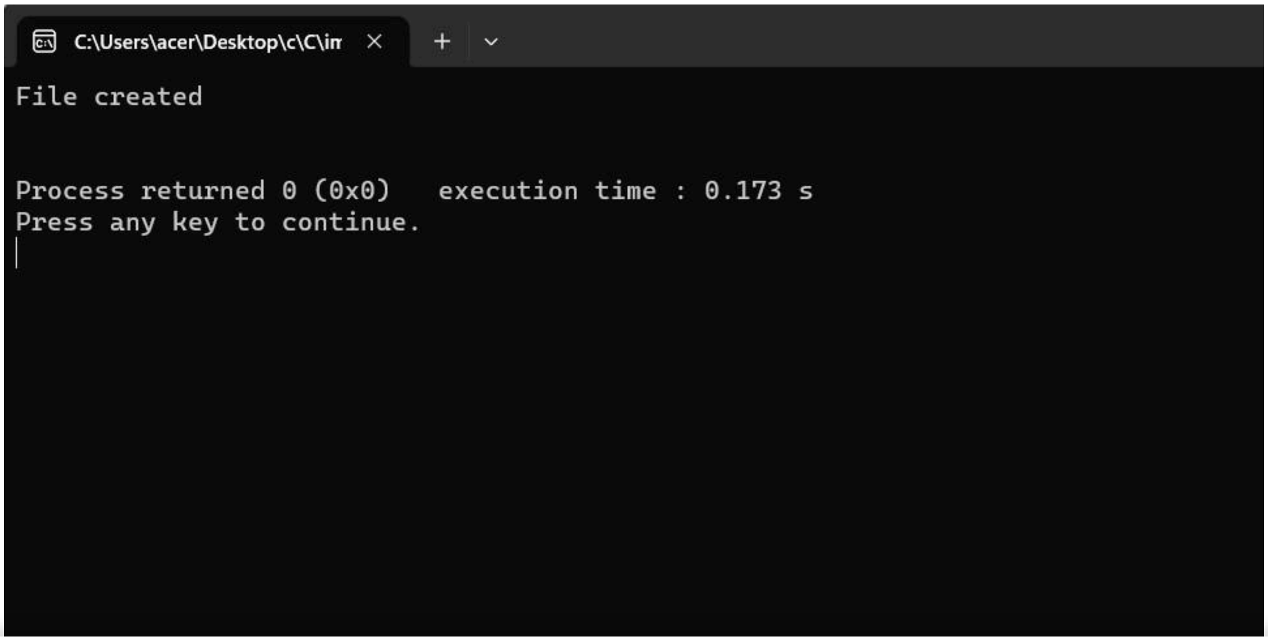
Creating a File in C

As now we know how to open a file using fopen() now the question arises about creation. The creation of a file is as simple as opening a file. As, if the while opening a file for writing or appending is done with either write("w") or append("a") mode then in the case where the file doesn't exist a new file is created.

Example:

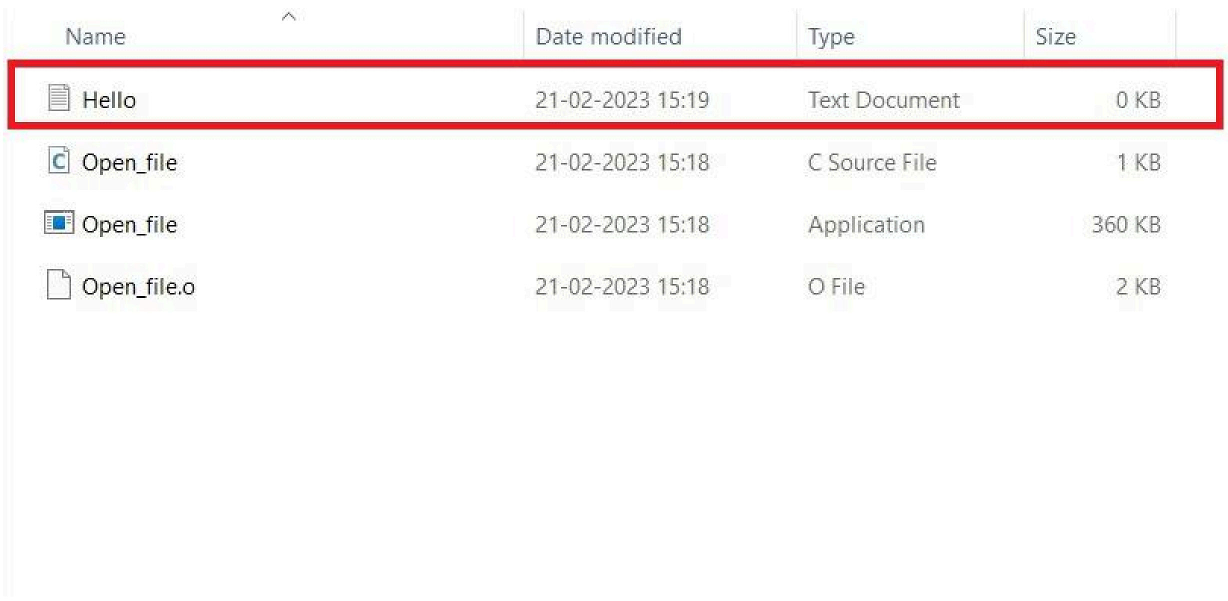
```
// C program to Create a file
// Driver code
int main()
{
    // File Pointer declared
    FILE* ptr;

      // File opened
    ptr = fopen("./Hello.txt", "w");

      // Failed Condition
    if (ptr == NULL) {
        printf("Error Occurred While creating a "
               "file !");
        exit(1);
    }

      // File closed
    fclose(ptr);

      // Data is finally Inserted
    printf("File created\n\n");

    return 0;
}
```

Output:



```
File created


Process returned 0 (0x0)   execution time : 0.173 s
Press any key to continue.
```

File Created:



Name	Date modified	Type	Size
Hello	21-02-2023 15:19	Text Document	0 KB
Open_file	21-02-2023 15:18	C Source File	1 KB
Open_file	21-02-2023 15:18	Application	360 KB
Open_file.o	21-02-2023 15:18	O File	2 KB

Hello.txt file created using the C program

Writing a File in C

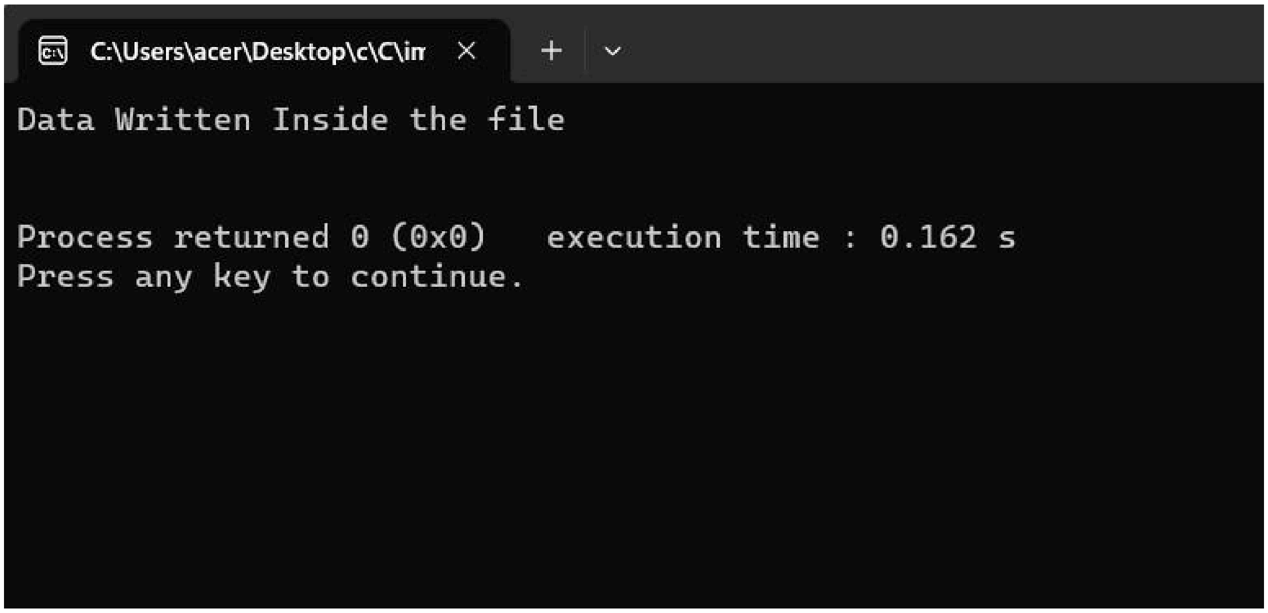
1. Writing to a text file in C

fprintf() and fscanf() are used to read and write in a text file in C programming. They expect a pointer to the structure FILE since they are file versions of print() and scanf().

```
// C program to write contents
// to the file
#include <stdio.h>
#include <stdlib.h>

// Driver code
int main()
{
    // File Pointer declared
    FILE* ptr;

      // File opened
    ptr = fopen("./Hello.txt", "w+");

      // Failed Condition
```

```
        exit(1);
    }

      // Data to be inserted
    char str[] = "This is all the Data to be inserted in "
                 "File by GFG.";

      // Puts data inside the file
    fputs(str, ptr);

      // File closed
    fclose(ptr);

      // Data is finally Inserted
    printf("Data Written Inside the file\n\n");

    return 0;
}
```

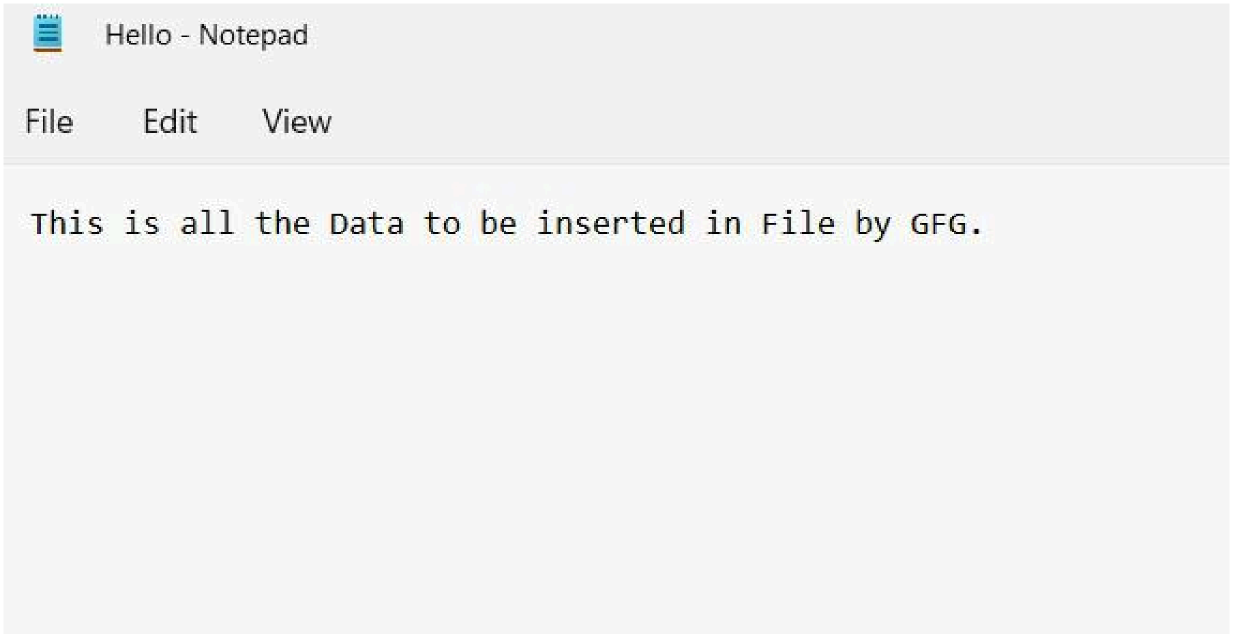
Output:



```
Data Written Inside the file


Process returned 0 (0x0)   execution time : 0.162 s
Press any key to continue.
```

The output of Writing to a text file in C

File Created:



```
This is all the Data to be inserted in File by GFG.
```

Writing to a text file in C

The above program takes a string from a user and stores it in text_file.text.After compiling this program a text file named temp_text.txt will be created in the C_Program folder. Inside the file, we can see the string that we entered.

2. Writing to a Binary File

fwrite() function takes four arguments address of data, size of the data which is to be written on the disk, number of data types, and a pointer to the file where we want to write.

Syntax:

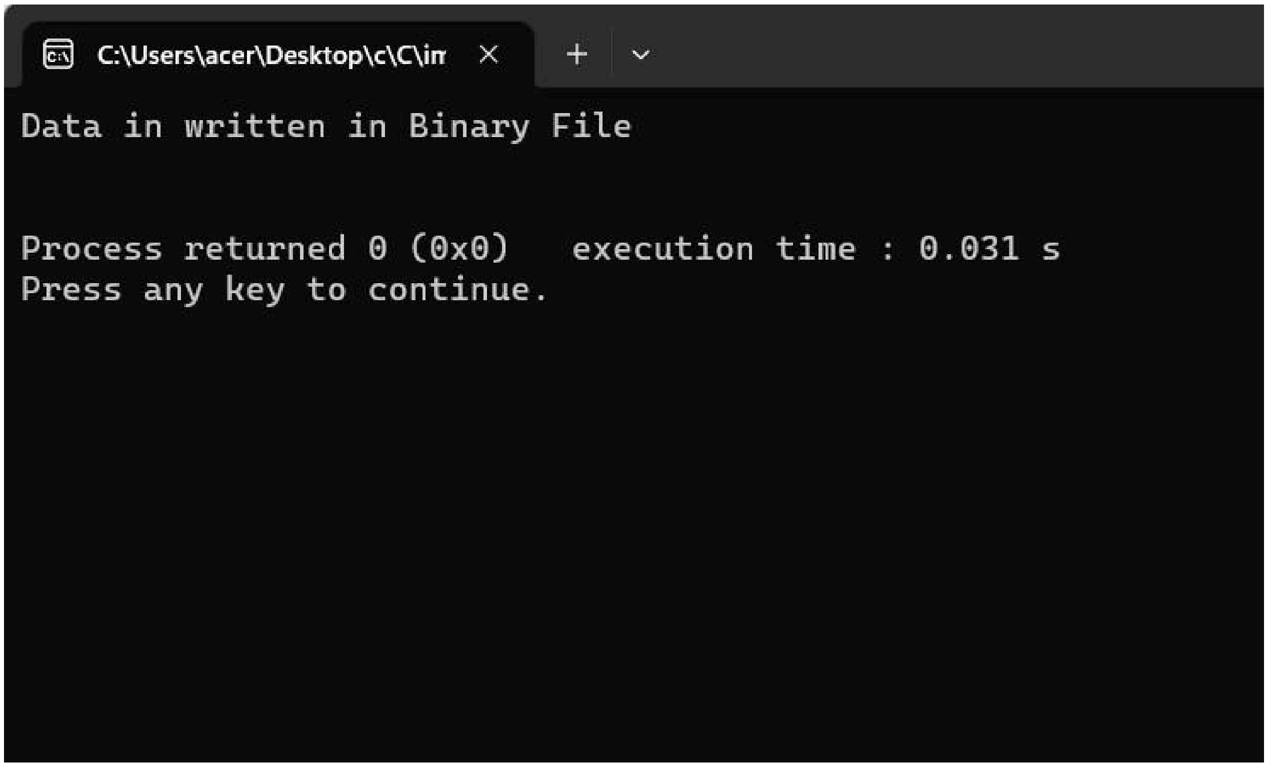
```
fwrite(const void *ptr,size_of_elements,number_of_elements, FILE
*a_file);
```

Below is the C program to write to a binary file:

```
// C program to write to a
// binary file
#include <stdio.h>
#include <stdlib.h>

// Struct declared
struct Num {
    int n1, n2;
};
```

```
      // variables declared
    int n;
    struct Num obj;

      // File Pointers declared
    FILE* fptr;

      // Failure Condition
    if ((fptr = fopen("temp.bin", "wb")) == NULL) {
        printf("Error! opening file");

        // if file pointer returns NULL program
        // will exit
        exit(1);
    }


    for (n = 1; n < 10; n++) {
        obj.n1 = n;
        obj.n2 = 12 + n;

      // Data written
        fwrite(&obj, sizeof(struct Num), 1, fptr);
    }

      // File closed
    fclose(fptr);

    printf("Data in written in Binary File\n\n");

    return 0;
}
```

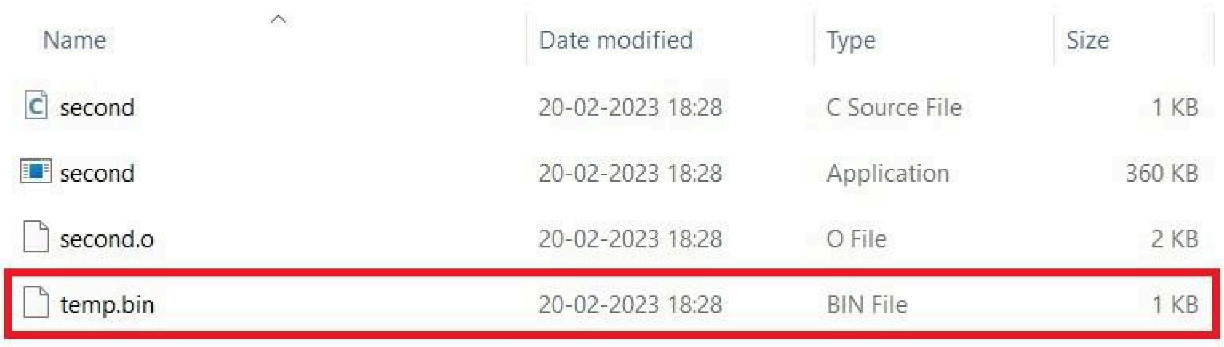
Output:



```
Data in written in Binary File


Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

The output of Writing to a Binary File

Image of created file:



Name	Date modified	Type	Size
second	20-02-2023 18:28	C Source File	1 KB
second	20-02-2023 18:28	Application	360 KB
second.o	20-02-2023 18:28	O File	2 KB
temp.bin	20-02-2023 18:28	BIN File	1 KB

.bin file created with the data

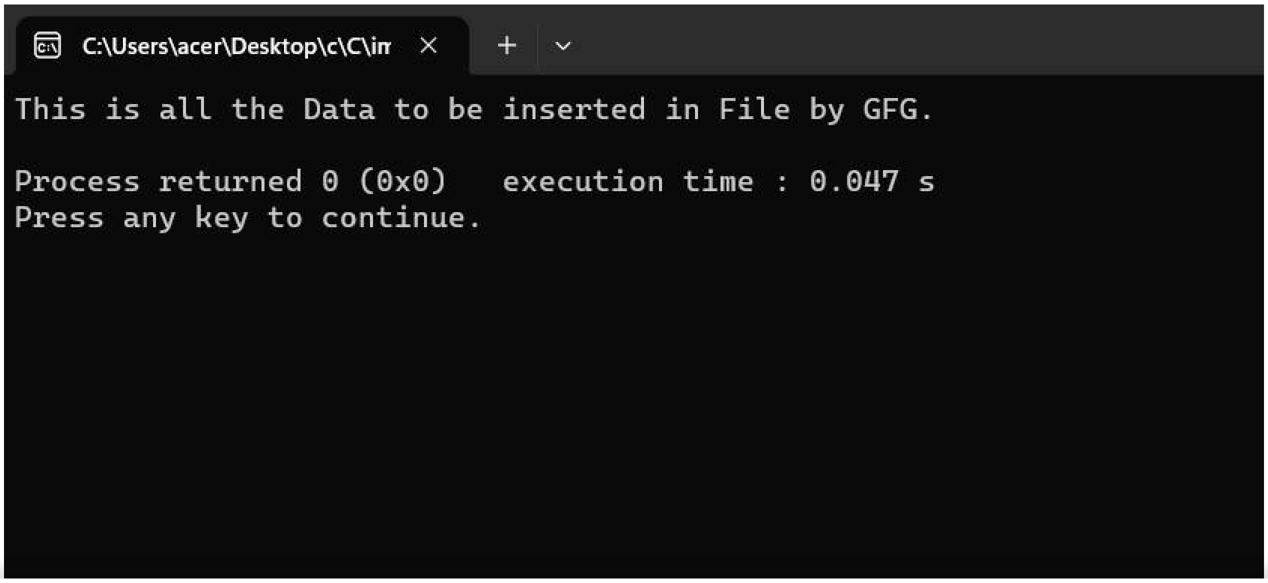
> ***Note:*** *fread() is used to read from a binary file and fwrite() is used to write to a file on the disk.*

Reading File in C

Below is the C program to read the contents from the file:

```c
// C program to read contents
// from the file
#include <stdio.h>
#include <stdlib.h>

// Driver code
int main()
{
  char str[80];
  FILE* ptr;

  ptr = fopen("Hello.txt", "r");

  if (ptr == NULL)
  {
    printf("Error While opening file");

    // if the pointer returns NULL
    // program will exit
    exit(1);
  }

  if(fgets(str, 80, ptr) != NULL)
  {
    puts(str);
  }
  fclose(ptr);

  return 0;
}
```

Output:



```
This is all the Data to be inserted in File by GFG.

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

2. Reading from a Binary File

fread() function also takes four arguments that are similar to fwrite() function in C Programming.

Syntax:

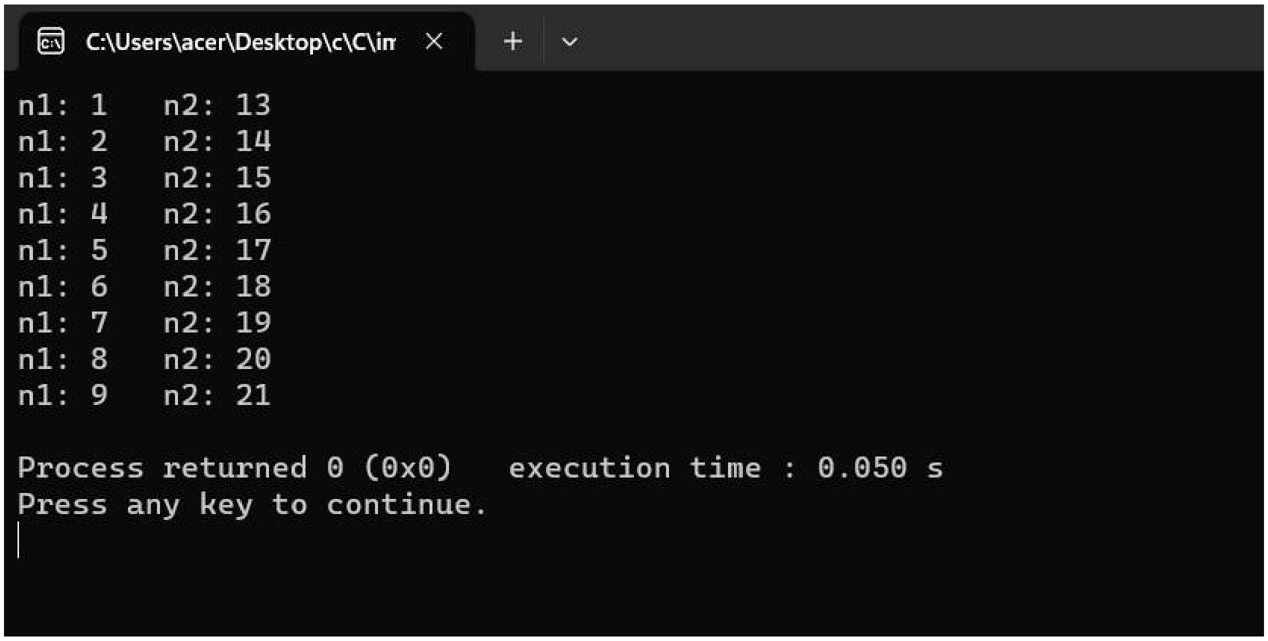
```
fwrite(const void *ptr,size_of_elements,number_of_elements, FILE
*a_file);
```

Below is the C program to read from a binary file:

```
// C Program to Read from a
// binary file using fread()
#include <stdio.h>
#include <stdlib.h>
struct Num
{
    int n1, n2;
};

// Driver code
int main()
{
  int n;
  struct Num obj;
  FILE* fptr;
  if ((fptr = fopen("temp.bin", "rb")) == NULL)
  {
    printf("Error! opening file");

    // If file pointer will return NULL
    // Program will exit.
    exit(1);
  }

  // else it will return a pointer
  // to the file.
  for (n = 1; n < 10; ++n)
  {
```

```
  fclose(fptr);
  return 0;
}
```

Output:



```
n1: 1   n2: 13
n1: 2   n2: 14
n1: 3   n2: 15
n1: 4   n2: 16
n1: 5   n2: 17
n1: 6   n2: 18
n1: 7   n2: 19
n1: 8   n2: 20
n1: 9   n2: 21

Process returned 0 (0x0)   execution time : 0.050 s
Press any key to continue.
```

The output of Reading from a Binary File

Explanation: In the above program, we have read the same file GFG.bin and are looping through records one by one. We read a single Num record of Num size from the file pointed by *fptr into the structure Num. We'll get the same record that we inserted in the previous program.

Moving File Pointers to Specific Positions

fseek() and rewind() are the two methods in C programming that can be used to move the file pointer. Let us check both methods:

1. fseek() in C Programming

fseek() function is used to set the file pointer to the specified offset and write data into the file.

Syntax:

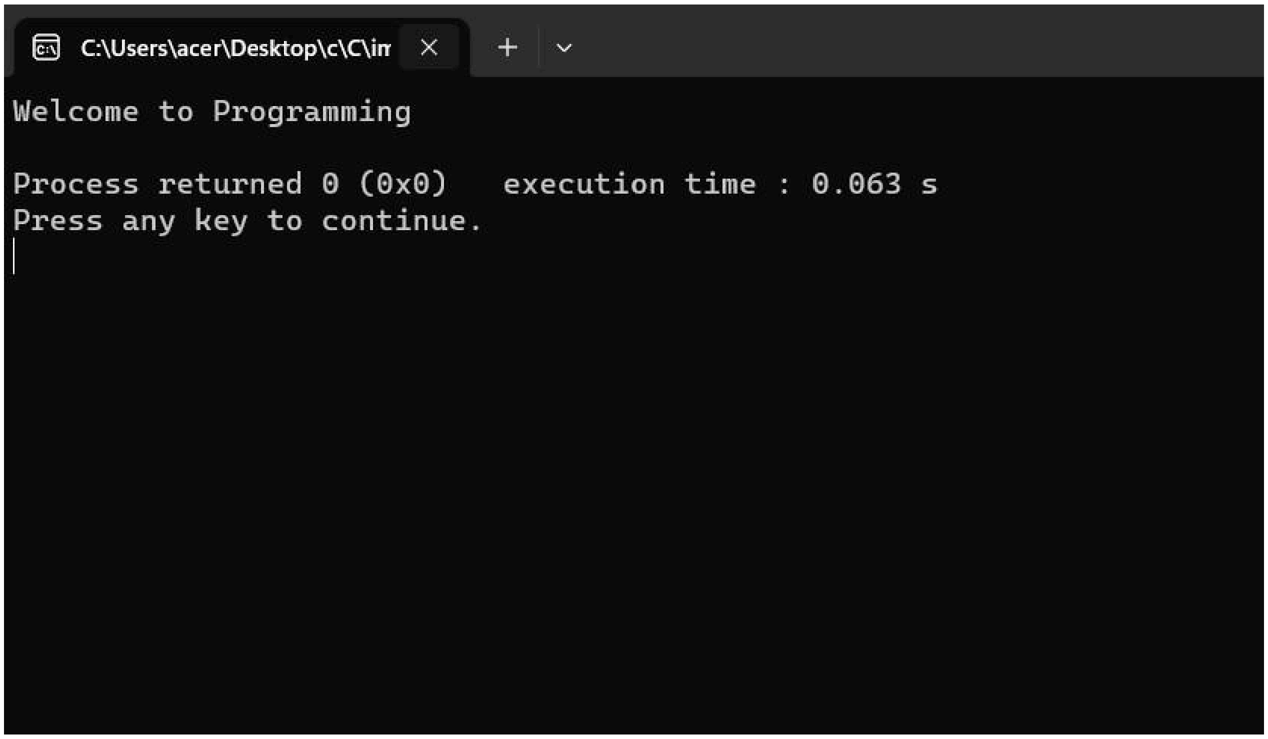
Here,

- **whence** can be SEEK_SET, SEEK_CUR and SEEK_END.
- **SEEK_END:** It denotes the end of the file.
- **SEEK_SET:** It denotes starting of the file.
- **SEEK_CUR:** It denotes the file pointer's current position.

Below is the C program to implement fseek():

```
// C program to implement
// fseek
#include <stdio.h>
#include <stdlib.h>

// Driver code
int main()
{
      // string declared
    char str[80];

      // File Pointer declared
    FILE* ptr;

      // File Opened
    ptr = fopen("Hello.txt", "w+");

      // Puts data in File
      fputs("Welcome to GeeksforGeeks", ptr);

      // fseek function used
    fseek(ptr, 11, SEEK_SET);

      // puts programming in place of at position defined
      // After 11 elements
    fputs("Programming  ", ptr);
    fclose(ptr);

      // Reading the file
    ptr = fopen("Hello.txt", "r+");
    if (fgets(str, 80, ptr) != NULL) {
      // Close the opened file
    fclose(ptr);

    return 0;
}
```

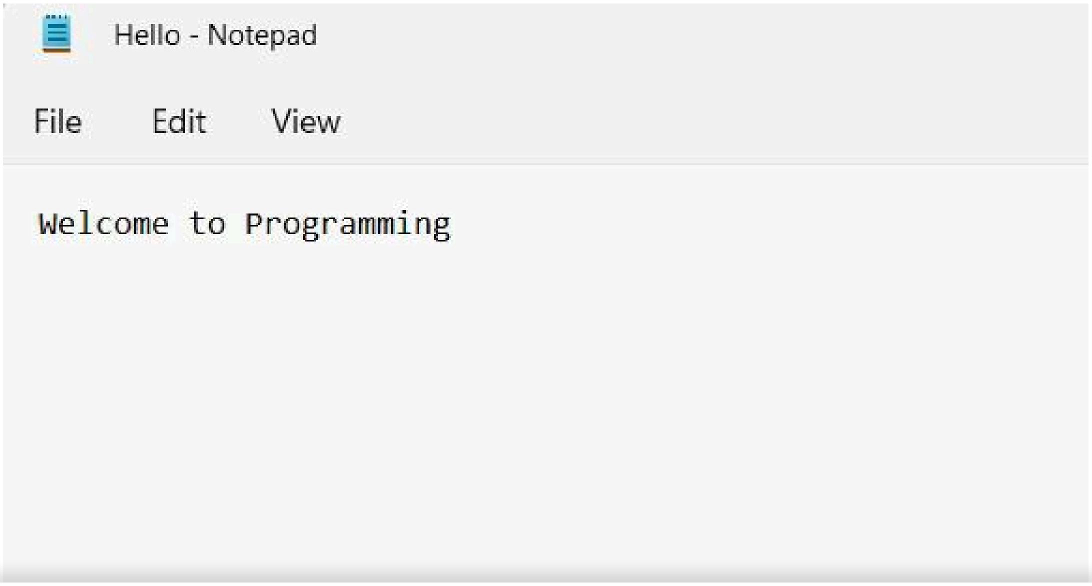
Output:



```
Welcome to Programming

Process returned 0 (0x0)   execution time : 0.063 s
Press any key to continue.
```

The output of fseek() in C

Image of the File:



```
Welcome to Programming
```

Changes in the file

2. rewind() in C

rewind() function sets the file pointer to the beginning of the file.

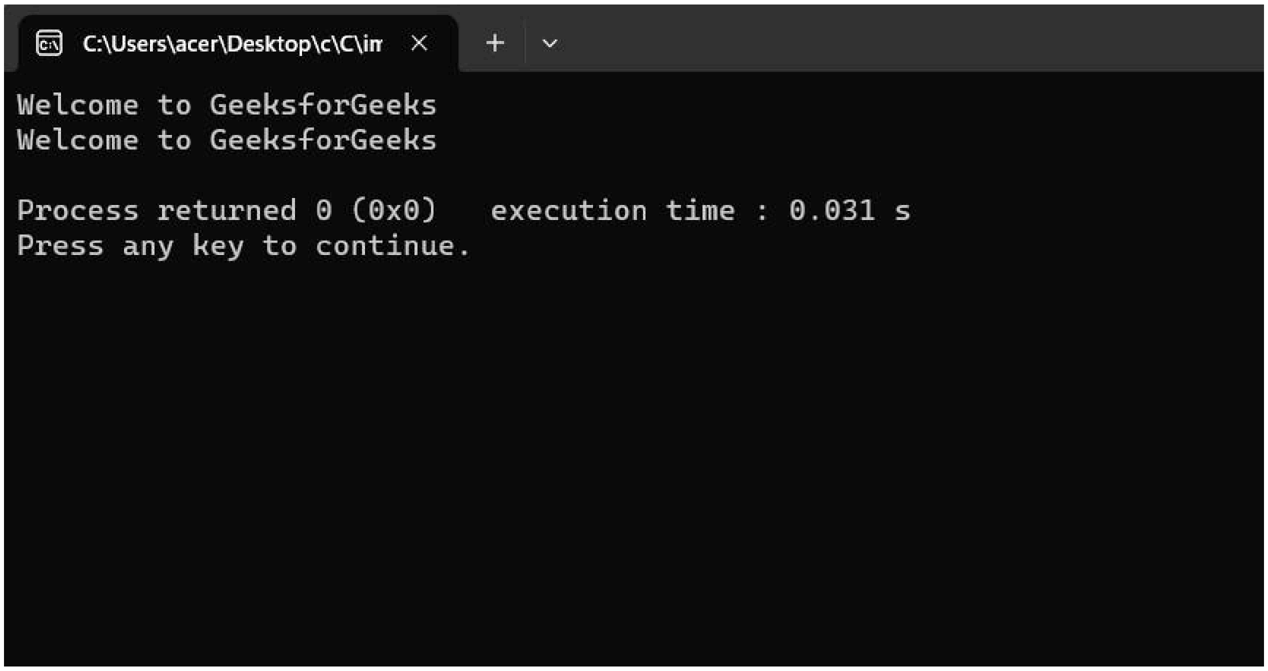
Syntax:

```
void rewind(FILE *stream);
```

Below is the C Program to implement rewind():

```
// C program to implement
// rewind()
#include <stdio.h>
#include <stdlib.h>

// Driver code
int main()
{
  // String declared
    char str[200];

      // File Pointer declared
    FILE* ptr;

      // File opened
    ptr = fopen("Hello.txt", "w+");

      // Puts data inside the file
    fputs("Welcome to GeeksforGeeks", ptr);

      // File closed
    fclose(ptr);

      // File open to read
    ptr = fopen("Hello.txt", "r+");
    if (fgets(str, 200, ptr) != NULL) {
        puts(str);
    }
    rewind(ptr);
    if (fgets(str, 200, ptr) != NULL) {
        puts(str);
    }

    fclose(ptr);
    return 0;
}
```

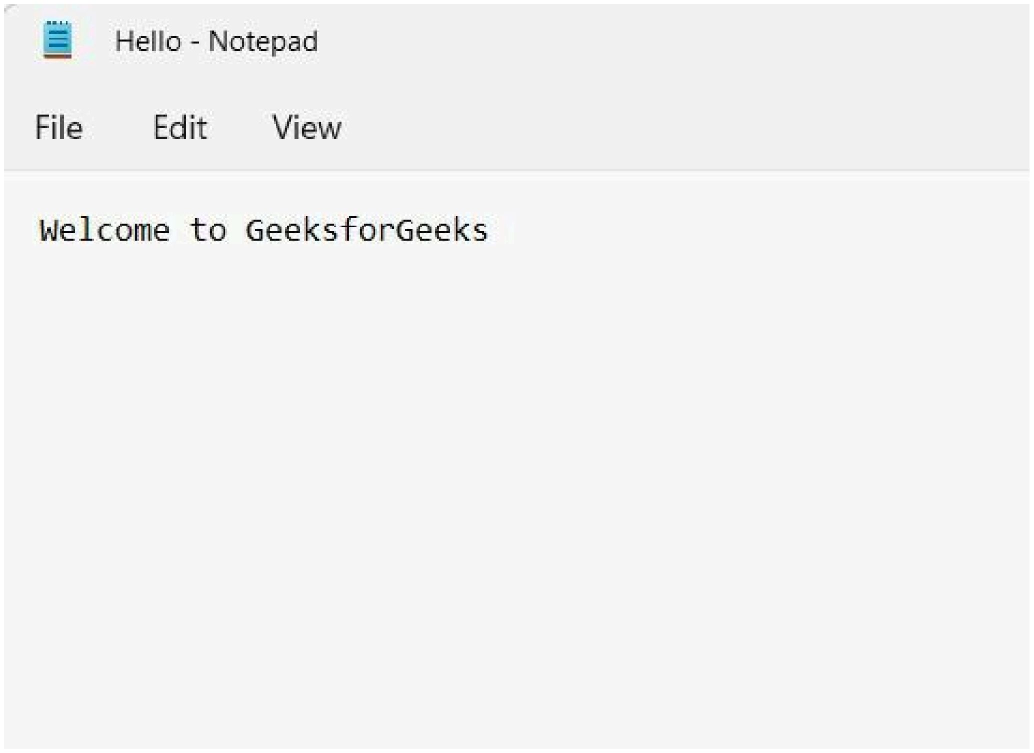
Output:



```
Welcome to GeeksforGeeks
Welcome to GeeksforGeeks

Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

The output of rewind() in C

Image of the File:



```
Welcome to GeeksforGeeks
```

File after running the program

Comment More info Advertise with us

Next Article

fwrite() in C

Similar Reads

C File Pointer

A file pointer is a variable that is used to refer to an opened file in a C program. The file pointer is actually a structure that stores the file data such as the file name, its location, mode, and the current position in the file....

3 min read

fwrite() in C

In C, fwrite() is a built-in function used to write a block of data from the program into a file. It can write arrays, structs, or other data to files but is especially designed to write binary data in binary files.Example:C#include...

3 min read

fseek() in C

The fseek() function is part of the C standard library that is used to move the file pointer associated with a given file to a specific location in the file. It provides random access capabilities, which allow the programmer...

2 min read

In C, fgets() is a built-in function defined in <stdio.h> header file. It reads the given number of characters of a line from the input stream and stores it into the specified string. This function stops reading the characters...

3 min read

fprintf() in C

fprintf is used to print content in file instead of stdout console. int fprintf(FILE *fptr, const char *str, ...); Example: C // C Program for the above approach #include<stdio.h> int main() { int i, n=2; char str[50]; //open...

1 min read

C fopen() Function

In C, the fopen() function is used to open a file in the specified mode. The function returns a file pointer (FILE *) which is used to perform further operations on the file, such as reading from or writing to it. If the file exists...

5 min read

Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Python Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics

GeeksforGeeks Videos

DSA
Python

English Grammar
Data Science
Commerce
CS Subjects
World GK