



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* M. I. Marco Antonio Martínez Quintana

*Asignatura:* Fundamentos de programación

*Grupo:* 3

*No de Práctica(s):* #13

*Integrante(s):* Fausto Ángel Reséndiz Álvarez

*No. de Equipo de  
cómputo empleado:* No aplica

*No. de Lista o Brigada:* 38

*Semestre:* 2021-1

*Fecha de entrega:* 22/01/2021

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_

## Objetivo

Elaborar programas en lenguaje C que requieran el uso de archivos de texto plano en la resolución de problemas, entendiendo a los archivos como un elemento de almacenamiento secundario.

## Introducción

Un conjunto de datos organizados para una colección de entidades de tipo básica o elemental son comúnmente clasificadas como registros. Estos son del mismo tipo, pertenecer a un mismo contexto y ser almacenados para su posterior uso de manera ordenada.

# Desarrollo

## ¿Cómo lo implementamos?

Vamos a escribir nuestros resultados del cálculo de la suma de los primeros números en un archivo.



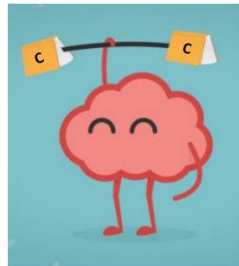
```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc gaussArchivos.c
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gaussArchivos.exe
Suma de los primeros n números
¿Cuántos números deseas sumar? 100
La suma de los primeros 100 números es: 5050
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gaussArchivos.exe
Suma de los primeros n números
¿Cuántos números deseas sumar? 10
La suma de los primeros 10 números es: 55
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gaussArchivos.exe
Suma de los primeros n números
¿Cuántos números deseas sumar? 120
La suma de los primeros 120 números es: 7260
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>

//Declarar variables
char au=163,sp=168,aa=160;
int n,res;
//Apuntador archivo
FILE *a;
a=fopen("ResultadosGauss.txt","w");
//Mensaje de bienvenida
printf("\n\nCálculo de la suma de los primeros n números\n\n",au);
//Solicitar el número de elementos a sumar
printf("%s¿Cuántos números deseas sumar? ",sp,aa,au);
scanf("%d",&n);
//sumar los n números
res=0;
for(int i=1;i<=n;i++)
{
    fprintf(a,"%d + %d = ",res, i);
    res=res+i;
    fprintf(a,"%d\n",res);
}
/*
n = 5
i = 6
res = 15
res = 10 + 5 = 15
*/
//Mostrar el resultado
printf("La suma de los primeros %d números es: %d\n",n,au,res);
fclose(a);
return 0;
```

## Ejercicio 7

- Crear un programa que escriba los pasos del cálculo del factorial de un número en un archivo llamado factorial.txt

$1! = 1$   
 $2! = 2(1) = 2$   
 $3! = 3(2)(1) = 6$   
 $4! = 4(3)(2)(1) = 24$   
 $5! = 5(4)(3)(2)(1) = 120$

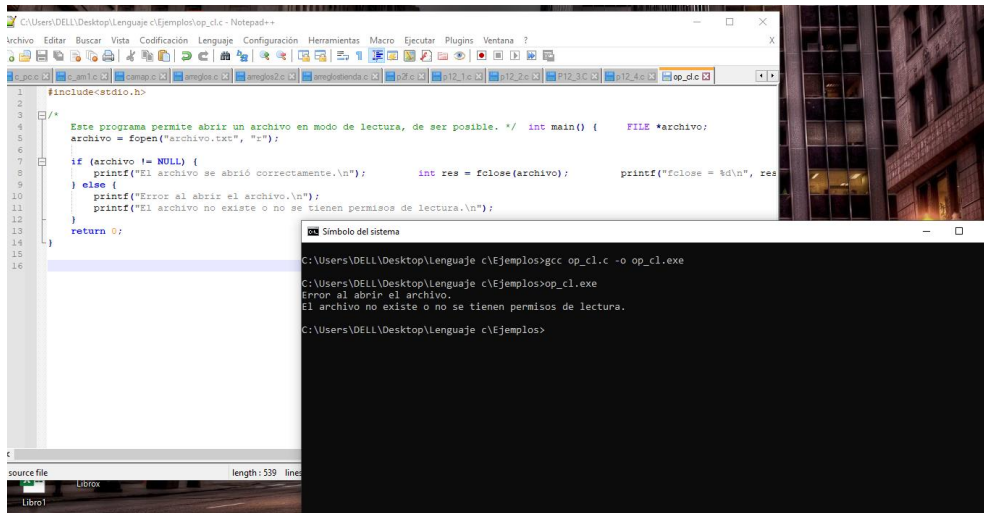


```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc factorialArchivos.c -o factorialArchivos.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>factorialArchivos.exe
Calculamos el factorial de un número
Escribe el valor que deseas calcular5
El factorial de 5 es: 120
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>

#include<stdio.h>
int main ()
{
    //Declarar variables
    int n, res, b;
    char au=163;
    //Apuntador archivo
    FILE *a;
    a=fopen("ResultadosFactorial.txt","w");
    //Mensaje de bienvenida
    printf("\n\nCálculo del factorial de un número\n\n",au);
    //Solicitar el número para calcular su factorial
    printf("%sEscribe el valor que deseas calcular:",sp,aa,au);
    scanf("%d",&n);
    //Realizar operación
    res=1;
    b=1;
    while(b<=n)
    {
        fprintf(a,"%d!= %d(%d) = ",b, b, res);
        res=res*b;
        b++;
        fprintf(a,"%d\n",res);
    }
    //Dar el resultado obtenido
    printf("El factorial de %d es: %d\n",n,res);
    fclose(a);
    return 0;
}
```

ResultadosFactorial: Bloc de notas  
Archivo Edición Formato Ver Ayuda  
1!= 1(1) = 1  
2!= 2(1) = 2  
3!= 3(2) = 6  
4!= 4(6) = 24  
5!= 5(24) = 120  
Línea 1, columna 1 100% Windows (CRLF) UTF-8

## Código (abrir cerrar archivo)



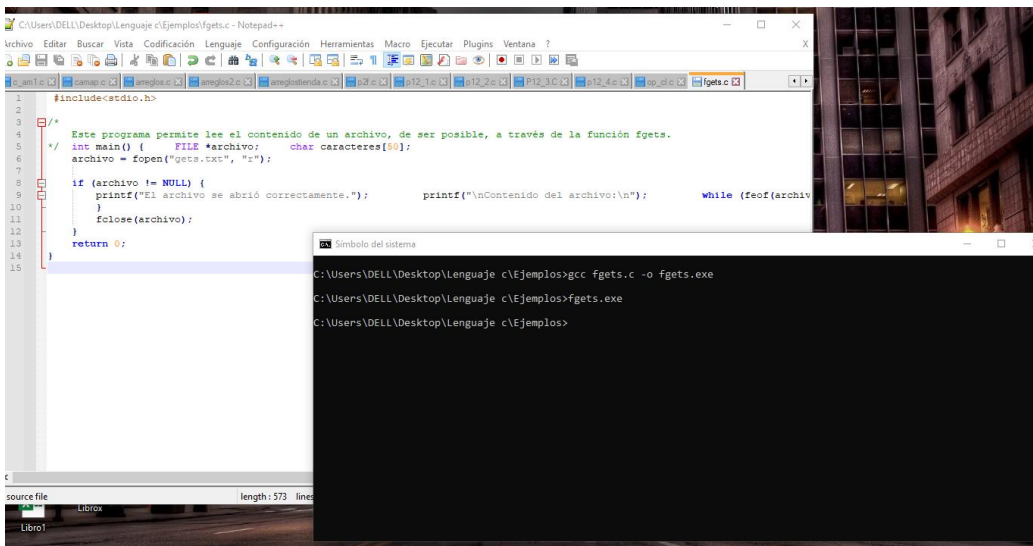
```
#include<stdio.h>

/*
Este programa permite abrir un archivo en modo de lectura, de ser posible. */
int main() { FILE *archivo;
archivo = fopen("archivo.txt", "r");

if (archivo != NULL) {
printf("El archivo se abrió correctamente.\n"); int res = fclose(archivo); printf("fclose = %d\n", res);
} else {
printf("Error al abrir el archivo.\n");
printf("El archivo no existe o no se tienen permisos de lectura.\n");
return 0;
}
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc op.cl.c -o op.cl.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>op.cl.exe
Error al abrir el archivo.
El archivo no existe o no se tienen permisos de lectura.
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

## Código (fgetc)



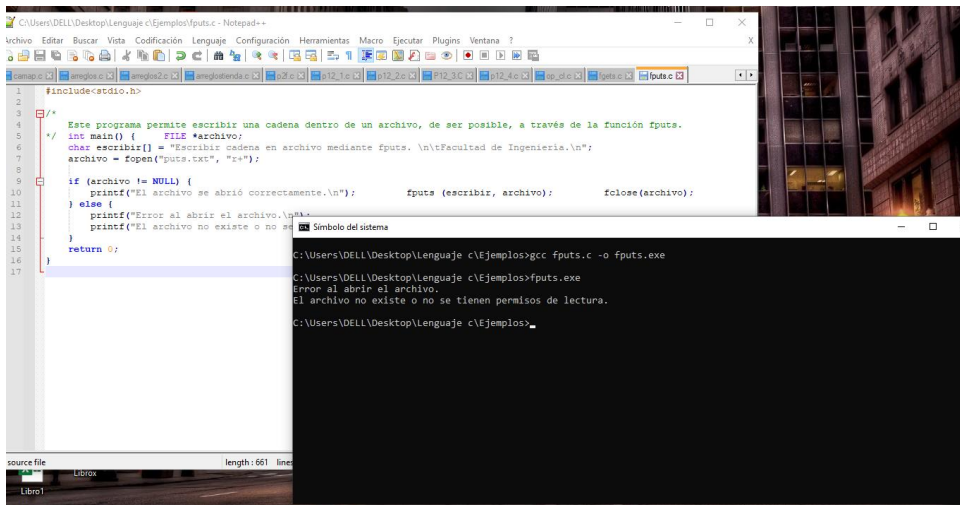
```
#include<stdio.h>

/*
Este programa permite leer el contenido de un archivo, de ser posible, a través de la función fgetc.
*/
int main() { FILE *archivo; char caracteres[90];
archivo = fopen("gets.txt", "r");

if (archivo != NULL) {
printf("El archivo se abrió correctamente."); printf("\nContenido del archivo:\n"); while (feof(archivo) != 1) {
char c = fgetc(archivo);
printf("%c", c);
}
fclose(archivo);
} else {
return 0;
}
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc fgetc.c -o fgetc.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>fgetc.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

## Código (fputs)



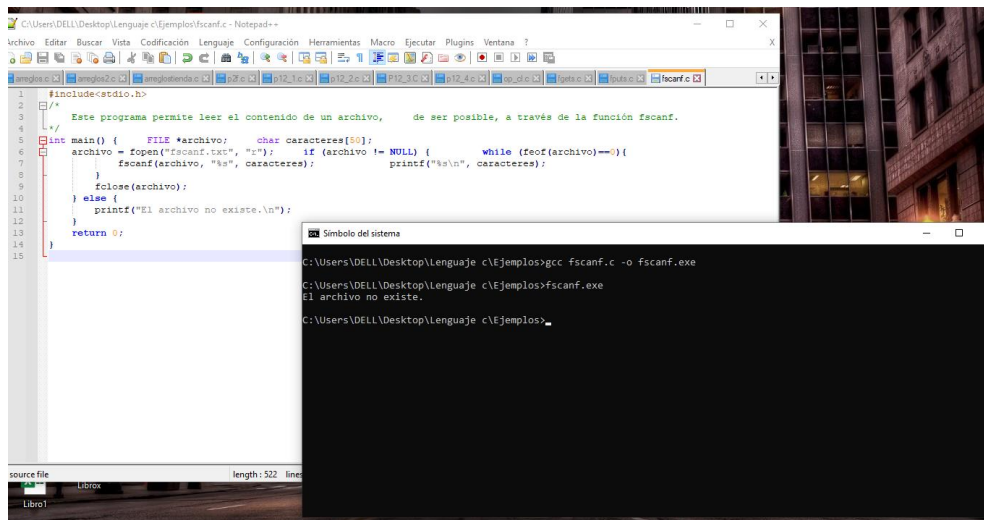
```
#include<stdio.h>

/*
Este programa permite escribir una cadena dentro de un archivo, de ser posible, a través de la función fputs.
*/
int main() { FILE *archivo;
char escribir[] = "Escribir cadena en archivo mediante fputs. \n\tFacultad de Ingenieria.\n";
archivo = fopen("puts.txt", "w");

if (archivo != NULL) {
printf("El archivo se abrió correctamente.\n"); fputs (escribir, archivo); fclose(archivo);
} else {
printf("Error al abrir el archivo.\n");
printf("El archivo no existe o no se tienen permisos de lectura.\n");
return 0;
}
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc fputs.c -o fputs.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>fputs.exe
Error al abrir el archivo.
El archivo no existe o no se tienen permisos de lectura.
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

## Código (fscanf)

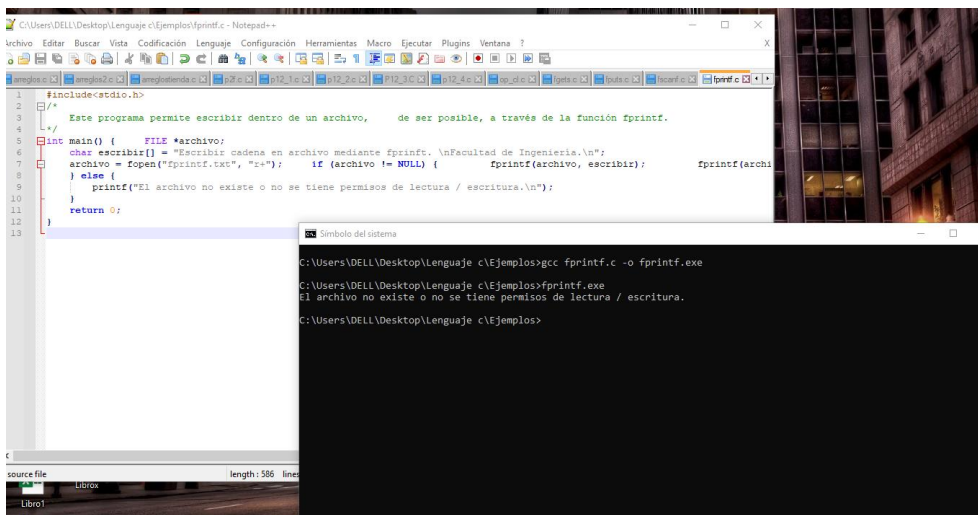


The screenshot shows a C program in Notepad++ and its execution in a Windows command prompt. The program, named `fscanf.c`, includes `<stdio.h>` and has a comment: "Este programa permite leer el contenido de un archivo, de ser posible, a través de la función fscanf." The `main` function attempts to open `"fscanf.txt"` in read mode. If the file is not null, it uses `fscanf` to read characters and prints them. If the file does not exist, it prints an error message. The command prompt shows the compilation command `gcc fscanf.c -o fscanf.exe` and the execution command `fscanf.exe`, which results in the output "El archivo no existe."

```
1 #include <stdio.h>
2
3 /*
4  * Este programa permite leer el contenido de un archivo, de ser posible, a través de la función fscanf.
5  */
6
7 int main() { FILE *archivo; char caracteres[50];
8     archivo = fopen("fscanf.txt", "r"); if (archivo != NULL) { while (feof(archivo)==0){
9         fscanf(archivo, "%s", caracteres); printf("%s\n", caracteres);
10    }
11    fclose(archivo);
12 } else { printf("El archivo no existe.\n");
13 }
14 return 0;
15 }
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc fscanf.c -o fscanf.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>fscanf.exe
El archivo no existe.
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

## Código (fprintf)

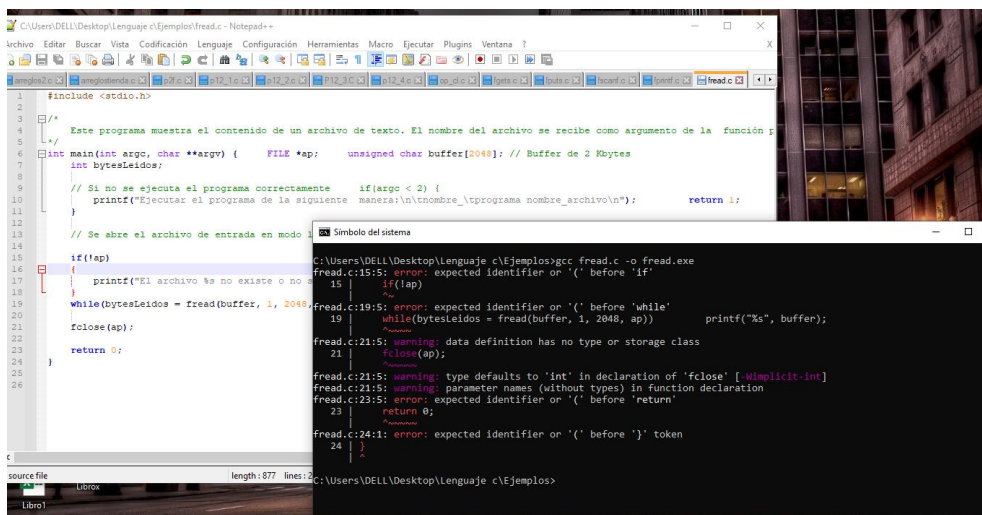


The screenshot shows a C program in Notepad++ and its execution in a Windows command prompt. The program, named `fprintf.c`, includes `<stdio.h>` and has a comment: "Este programa permite escribir dentro de un archivo, de ser posible, a través de la función fprintf." The `main` function attempts to open `"fprintf.txt"` in write mode. If the file is not null, it uses `fprintf` to write a string to the file. If the file does not exist or permissions are lacking, it prints an error message. The command prompt shows the compilation command `gcc fprintf.c -o fprintf.exe` and the execution command `fprintf.exe`, which results in the output "El archivo no existe o no se tiene permisos de lectura / escritura."

```
1 #include <stdio.h>
2
3 /*
4  * Este programa permite escribir dentro de un archivo, de ser posible, a través de la función fprintf.
5  */
6
7 int main() { FILE *archivo;
8     char escribir[] = "Escribir cadena en archivo mediante fprintf. \nFacultad de Ingenieria.\n";
9     archivo = fopen("fprintf.txt", "w"); if (archivo != NULL) { fprintf(archivo, escribir);
10 } else { printf("El archivo no existe o no se tiene permisos de lectura / escritura.\n");
11 }
12 return 0;
13 }
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc fprintf.c -o fprintf.exe
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>fprintf.exe
El archivo no existe o no se tiene permisos de lectura / escritura.
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>
```

## Código (fread)



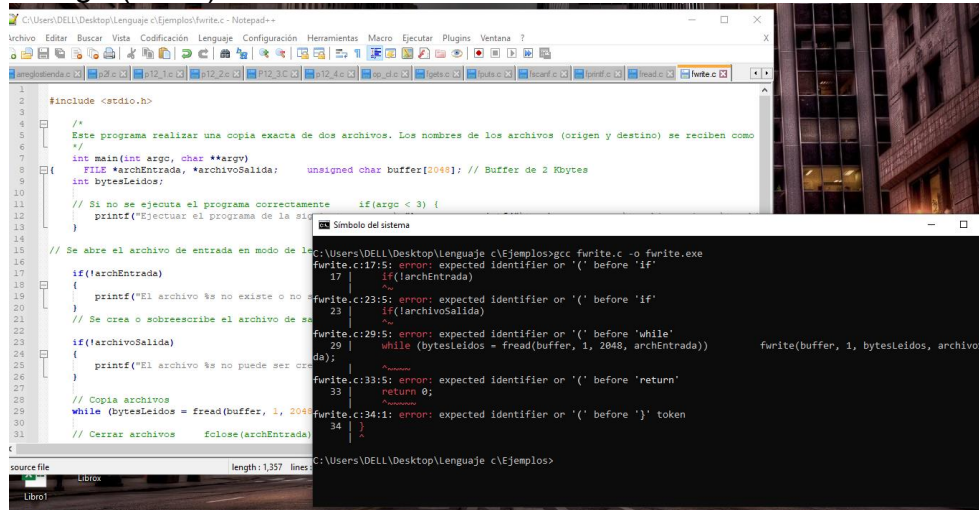
The screenshot shows a C program in Notepad++ and its compilation in a Windows command prompt. The program, named `fread.c`, includes `<stdio.h>` and has a comment: "Este programa muestra el contenido de un archivo de texto. El nombre del archivo se recibe como argumento de la función fread." The `main` function takes command-line arguments and attempts to open a file. It uses `fread` to read data from the file and prints it. The command prompt shows the compilation command `gcc fread.c -o fread.exe`, which results in several error messages. The errors include: "error: expected identifier or '(' before 'if'", "error: expected identifier or '(' before 'while'", "warning: data definition has no type or storage class", "warning: type defaults to 'int' in declaration of 'fclose' [implicit-int]", "warning: parameter names (without types) in function declaration", "error: expected identifier or '(' before 'return'", and "error: expected identifier or '(' before '}' token".

```
1 #include <stdio.h>
2
3 /*
4  * Este programa muestra el contenido de un archivo de texto. El nombre del archivo se recibe como argumento de la función fread.
5  */
6
7 int main(int argc, char **argv) { FILE *ap; unsigned char buffer[2048]; // Buffer de 2 Kbytes
8     int bytesLeidos;
9
10    // Si no se ejecuta el programa correctamente if(argc < 2) {
11        printf("Ejecutar el programa de la siguiente manera:\n\nnombre_programa nombre_archivo\n"); return 1;
12    }
13
14    // Se abre el archivo de entrada en modo lectura
15    if(!ap)
16        printf("El archivo %s no existe o no se puede abrir.\n", argv[1]);
17
18    while(bytesLeidos = fread(buffer, 1, 2048, ap))
19        printf("%s", buffer);
20    fclose(ap);
21    return 0;
22 }
```

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc fread.c -o fread.exe
fread.c:15:5: error: expected identifier or '(' before 'if'
15 |     if(!ap)
    |     ^
fread.c:19:5: error: expected identifier or '(' before 'while'
19 |     while(bytesLeidos = fread(buffer, 1, 2048, ap))
    |     ^
fread.c:21:5: warning: data definition has no type or storage class
21 |     fclose(ap);
    |     ^~~~~~
fread.c:21:5: warning: type defaults to 'int' in declaration of 'fclose' [implicit-int]
fread.c:21:5: warning: parameter names (without types) in function declaration
fread.c:23:5: error: expected identifier or '(' before 'return'
23 |     return 0;
    |     ^~~~~~
fread.c:24:11: error: expected identifier or '(' before '}' token
24 | }
```

- Código con error

## Código (fwrite)



The image shows a Notepad++ window with a C program and a Windows command prompt window displaying compilation errors.

**Notepad++ Window:**

```
1 #include <stdio.h>
2
3
4 /*
5  * Este programa realizar una copia exacta de dos archivos. Los nombres de los archivos (origen y destino) se reciben como
6  */
7
8 int main(int argc, char **argv)
9 {
10     FILE *archEntrada, *archivoSalida;    unsigned char buffer[2048]; // Buffer de 2 Kbytes
11     int bytesLeidos;
12
13     // Si no se ejecuta el programa correctamente    if(argc < 3) {
14         printf("Ejecutar el programa de la siguiente manera: fwrite.exe <archivoEntrada> <archivoSalida>\n");
15     }
16
17     // Se abre el archivo de entrada en modo de lectura
18     if(!archEntrada)
19     {
20         printf("El archivo %s no existe o no se puede abrir\n", argv[1]);
21         // Se crea o sobrescribe el archivo de salida
22         if(!archivoSalida)
23         {
24             printf("El archivo %s no puede ser creado\n", argv[2]);
25         }
26     }
27     // Copia archivos
28     while (bytesLeidos = fread(buffer, 1, 2048, archEntrada))
29     {
30         // Cerrar archivos    fclose(archEntrada);
31     }
```

**Windows Command Prompt Window:**

```
C:\Users\DELL\Desktop\Lenguaje c\Ejemplos>gcc fwrite.c -o fwrite.exe
fwrite.c:17:5: error: expected identifier or '(' before 'if'
   17 |     if(!archEntrada)
      |     ~^
fwrite.c:23:5: error: expected identifier or '(' before 'if'
   23 |     if(!archivoSalida)
      |     ~^
fwrite.c:29:5: error: expected identifier or '(' before 'while'
   29 |     while (bytesLeidos = fread(buffer, 1, 2048, archEntrada))
      |     ~^
fwrite.c:33:5: error: expected identifier or '(' before 'return'
   33 |     return 0;
      |     ~^
fwrite.c:34:1: error: expected identifier or '(' before '}' token
   34 | }
      | ^
```

- Código con error

## Conclusión

Durante el desarrollo de esta práctica, en la cual abordamos temas sobre la lectura y escritura de datos, pudimos pasar de la teoría a la práctica guiándonos de primera instancia con un ejemplo en el cual creamos paso a paso un código donde al ejecutarlo y realizar su función, al mismo tiempo generaba un archivo de tipo "txt." Archivo en el cual se declaraban las acciones que realizaba el código al ejecutarse. Como segunda actividad se nos encomendó realizar un proceso similar, con una pequeña diferencia en la operación que realizaría el código, conservando la generación de un archivo "txt."

De igual forma obtuvimos información acerca del hilo apuntador archivo, las funciones abrir y cerrar archivo, entre otras. Con lo cual podemos concluir que la lectura y escritura de datos como una gran herramienta de apoyo para un programador, como un ejemplo podemos ver a los archivos "txt." Como una especie de almacenamiento secundario que nos permite guardar y documentar la ejecución y funcionamiento del proyecto que se está realizando.