|  |  |  |
| --- | --- | --- |
| Imagen que contiene dibujo, taza  Descripción generada automáticamente | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Ingeniería Karina García Morales |
| *Asignatura:* | Fundamentos de Programación |
| *Grupo:* | 20 |
| *No de Práctica:* | *12 Lectura y escritura de datosº* |
| *Integrante:* | Yushua Haza Jibaja |
| *No. de Equipo de cómputo empleado:* |  |
| *No. de Lista o Brigada:* | 22 |
| *Semestre:* | 2022 - 1 |
| *Fecha de entrega:* | 9 de diciembre de 2021 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Objetivos**

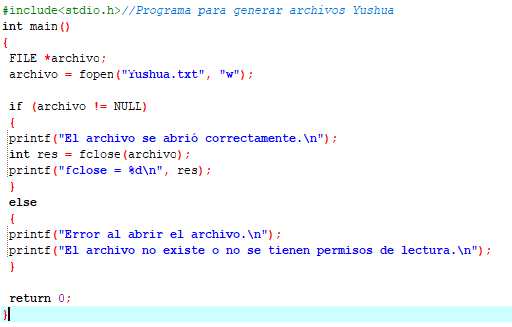
El alumno elaborará programas en C donde la solución del problema se divida en

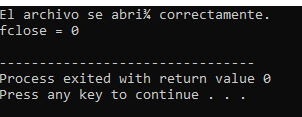
funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.

Programa1#

Se solicitó usar el código proporcionado para poder crear un archivo (Programa 1 de la práctica)

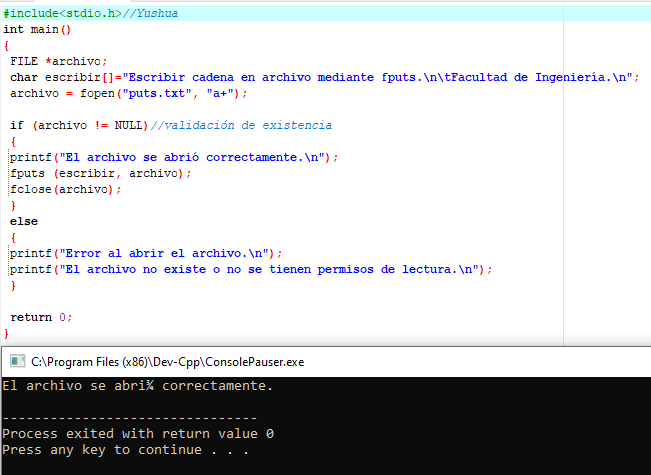
Se tuvo que cambiar el modo de lectura a escritura (“r” a “w”)

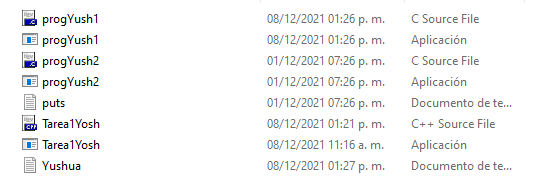




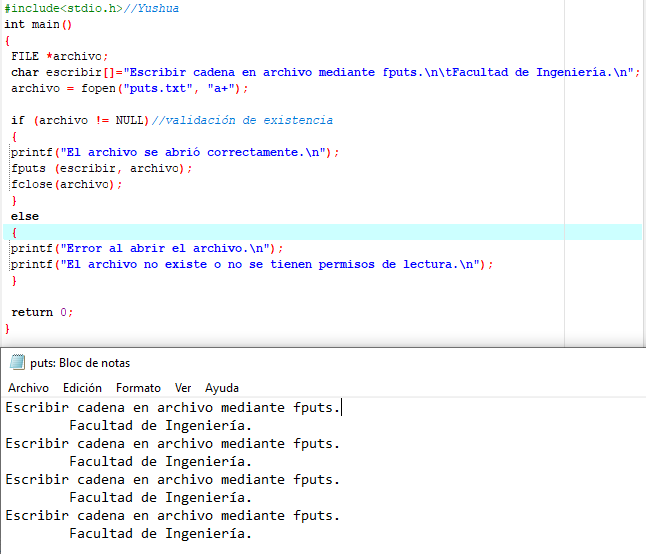
Programa 2#

Ejercicio de Fgets Fputs



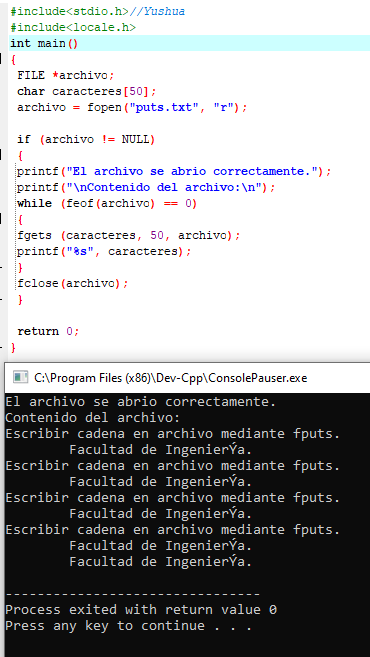


Y como el programa esta en modo de añadir, este toma el archivo de texto ya existente y agrega la misma línea de texto por cada vez que se ejecute el programa

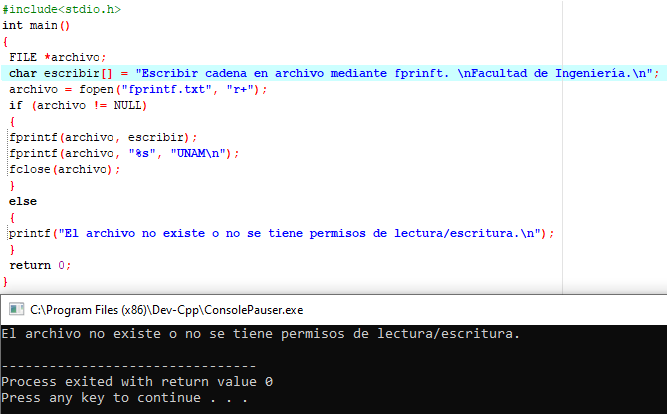


Programa 3#

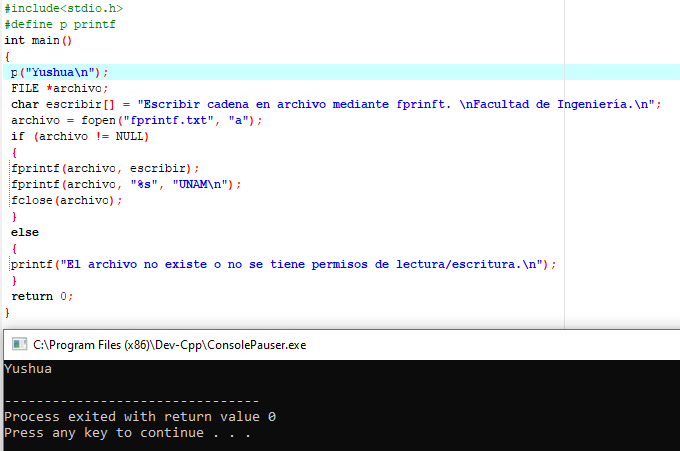
Tomando el mismo programa previo, modificamos el modo para que escriba una cadena corta



Programa 5 creo fprintf

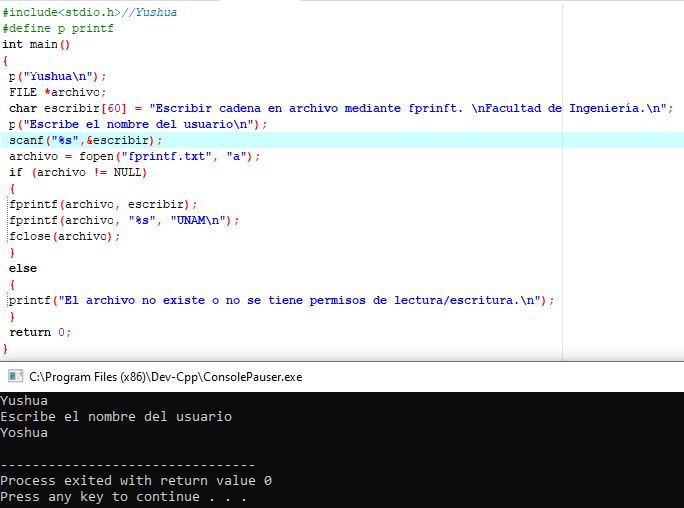


Cambiado el modo de “r+” a “a”

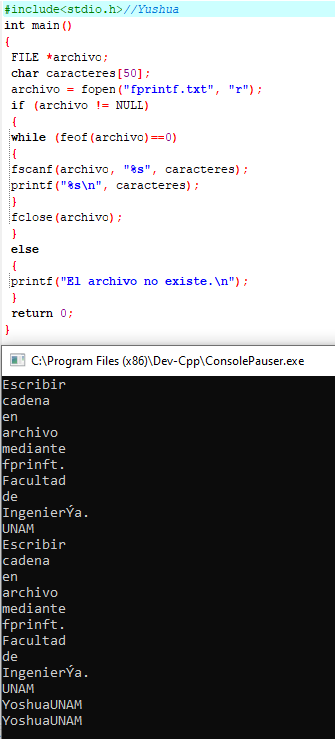


Programa 4#

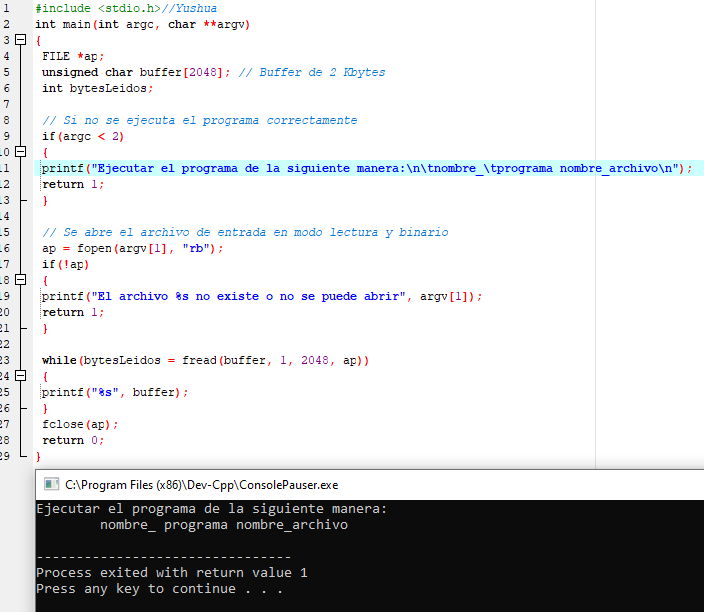
Es el mismo programa anterior pero ahora solicitando que el usuario ingrese un nombre



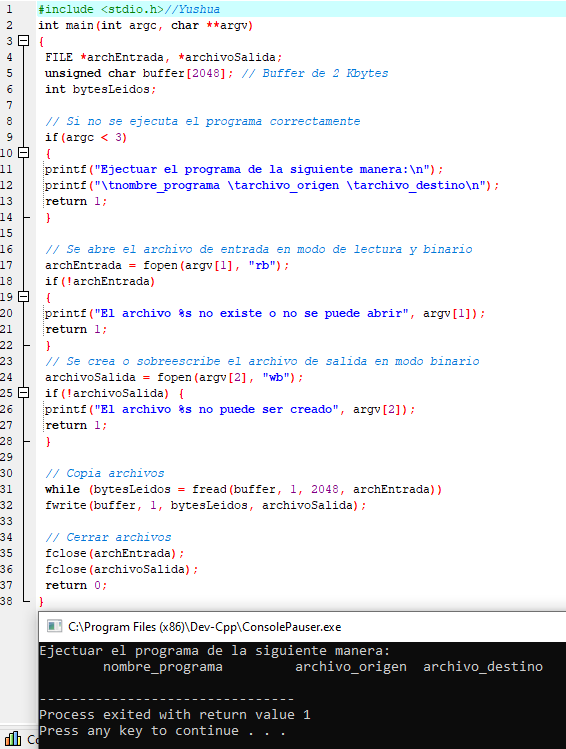
Programa 5# fscanf



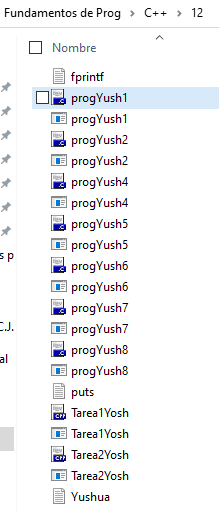
Programa #6



Programa #7



Prueba de archivos



Conclusión

Se pudo aprender exitosamente los comandos fputs y fgets para poder manipular la escritura en archivos de texto generados por un programa y como los modos de estos comandos afectan la misma escritura de caracteres en los archivos de texto simple.

Link de github: https://github.com/YushuaHaza/Practica-12

**Referencia**

-Facultad de Ingenieria. (2021). Laboratorio de computacion Sala A y B. Recuperado el 8 de diciembre de 2021 de <http://lcp02.fi-b.unam.mx/>

Tarea:

1.-Modifica los siguientes programas para emplear el uso de archivos almacenando los datos que ingresa el usuario y el resultado esperado:

----------Programa 1---------------

#include <stdio.h>

int main (){

char listaNombre[30],a;

int i, n;

printf("\n\n\tIngresa el nombre del alumno:");

gets(listaNombre);

for (i=0;i<=25;i++)

{

a=listaNombre[i];

if(a=='1'||a=='2'||a=='3'||a=='4'||a=='5'||a=='6'||a=='7'||a=='8'||a=='9'||a=='0')

{

n=i;

}

else

{

if(i!=n)

printf("%c",listaNombre[i]);

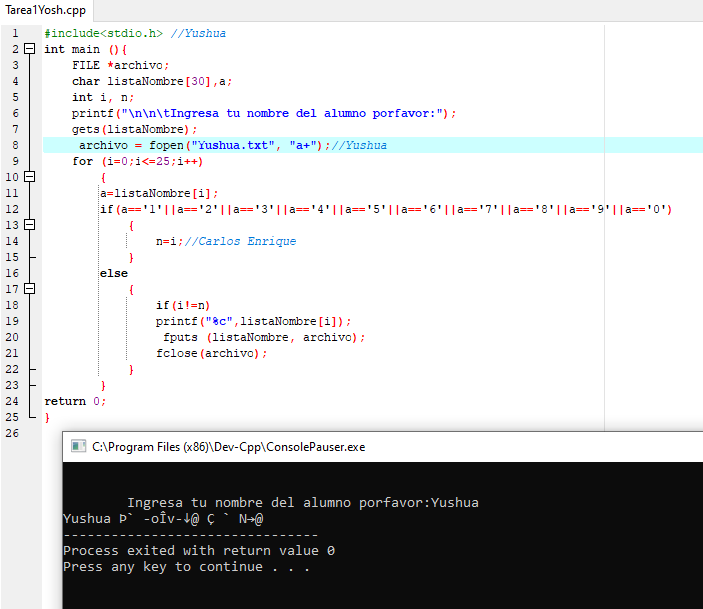
}

}

return 0;

}

Resolución



----------Programa 2---------------

#include <stdio.h>

#define p printf

#define s scanf

int b,c,d,dia,mes;

int fn\_anio(){

if(c>=1)

p("%d",c);

}

int fn\_diafeb()

{

if ( dia >= 1 && dia <= 28 )

{

printf( "%d",dia );

p(" de Febrero de ");

fn\_anio();

}else

printf( "\n\tFECHA INCORRECTA" );

}

int fn\_menu(int b)

{

if(b>=1||b<=12){

switch(b)

{

case 2:

fn\_diafeb();

break;

case 1:

p(" de Enero del ");

break;

case 3:

p(" de Marzo del ");

break;

case 5:

p(" de Mayo del ");

break;

case 7:

p(" de Julio del ");

break;

case 8:

p(" de Agosto del ");

break;

case 10:

p(" de Octubre del ");

break;

case 12:

p(" de Diciembre del ");

break;

case 4:

p(" de Abril del ");

break;

case 6:

p(" de Junio del ");

break;

case 9:

p(" de Septiembre del ");

break;

case 11:

p(" de Noviembre del ");

break;

default:

p("error");

break;

}

}else

p("error");

}

int fn\_fecha(int dia, int mes){

if (mes ==1||mes==3||mes ==5||mes==7||mes ==8||mes==10||mes ==12)

{

if( dia >=1&&dia <= 31 )

{

printf( "%d",dia);

fn\_menu(mes);

fn\_anio();

}else

printf( "\n FECHA INCORRECTA" );

}else if (mes ==4||mes==6||mes ==9||mes==11)

{

if(dia >=1&&dia<= 30 )

{

printf( "%d",dia);

fn\_menu(mes);

fn\_anio();

}

}else if (mes ==2)

{

if(dia<= 30 )

{

fn\_menu(mes);

}else

printf( "\n FECHA INCORRECTA" );

}

}

main()

{

p("Ingresa tres valores\n");

s("%d %d %d",&dia,&mes,&c);

fn\_fecha(dia,mes);

return 0;

}

**Codificación**

#include <stdio.h>//Yushua

#include<string.h>

#include<locale.h>

#include <stdlib.h>

#define p printf

#define fpf fprintf

#define s scanf

int a,b,c;

int dia;

int mes;

FILE \*archivo;

int amo(){

if(b>=1){p("%d",b); fpf(archivo, "%d", b);}

}

int diafeb()

{

if ( dia >= 1 && dia <= 28 )

{

p( "%d",dia ); fpf(archivo, "%d",dia);

p(" de Febrero de "); fpf(archivo, " de Febrero de ");

amo();

}else p( "\n\tFecha invalida" ); fpf(archivo, "\n\tFecha invalida" );

}

int menu(int a)

{

if(a>=1||a<=12){

switch(a)

{

case 1:

p(" de enero del "); fpf(archivo, " de Enero del ");

break;

case 2:

p("de febrero del "); fpf(archivo," de febrero del ");

case 3:

p(" de marzo del "); fpf(archivo, " de Marzo del ");

break;

case 4:

p(" de abril del "); fpf(archivo, " de Abril del ");

break;

case 5:

p(" de mayo del "); fpf(archivo, " de Mayo del ");

break;

case 6:

p(" de junio del "); fpf(archivo, " de Junio del ");

break;

case 7:

p(" de julio del "); fpf(archivo, " de Julio del ");

break;

case 8:

p(" de agosto del "); fpf(archivo, " de Agosto del ");

break;

case 9:

p(" de septiembre del "); fpf(archivo, " de Septiembre del ");

break;

case 10:

p(" de octubre del "); fpf(archivo, " de Octubre del ");

break;

case 11:

p(" de noviembre del "); fpf(archivo, " de Noviembre del ");

break;

case 12:

p(" de diciembre del "); fpf(archivo, " de Diciembre del ");

break;

default:

p("Error");

break;

}

}else

p("Error");

}//Yushua

int fecha(int dia, int mes){

if (mes ==1||mes==3||mes ==5||mes==7||mes ==8||mes==10||mes ==12)

{

if( dia >=1&&dia <= 31 )

{

p( "%d",dia); fpf(archivo, "%d",dia);

menu(mes);

amo();

}else

p( "\n Fecha invalida" );

}else if (mes ==4||mes==6||mes ==9||mes==11)

{

if(dia >=1&&dia<= 30 )

{

p( "%d",dia); fpf(archivo, "%d",dia);

menu(mes);

amo();

}

}else if (mes ==2)

{

if(dia<= 30 )

{

menu(mes);

}else

p( "\nFecha invalida" );

}

}

int main(){

printf ("Facultad de Ingeniería \n");

p("Ingresa num de dia, num de mes y año\n");

s("%d %d %d",&dia,&mes,&b);

archivo = fopen("YushuaTarea2.txt", "a+");

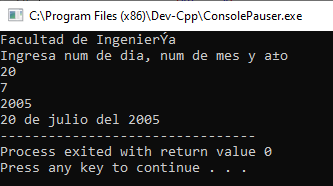
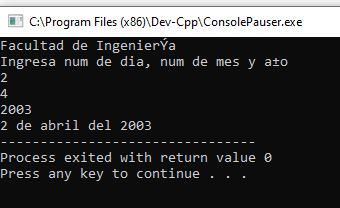
fecha(dia,mes);

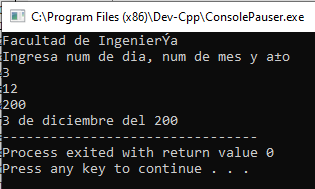
fclose(archivo);

return 0;

}

Pruebas de ejecución





2.- Completa el cuadro referente a funciones, sintaxis, ejemplo y características de cada una de las funciones vistas en el laboratorio.

Función Sintaxis Características Ejemplo de sintaxis

fputs char \*fputs(char \*buffer, FILE \*apArch); La función fputs() permite escribir una cadena en un archivo especifico fputs (escribir, archivo);

|  |  |  |  |
| --- | --- | --- | --- |
| Función | Sintaxis | Características | Ejemplo de sintaxis |
| fputs | char \***fputs**(char \*buffer, FILE \*apArch); | La función fputs() permite escribir una cadena en un archivo especifico | fputs (escribir archivo) |
| fclose | int **fclose**(FILE \*fichero); | cierra una secuencia que fue abierta mediante un fopen(). Escribe la información que se encuentre en el buffer al disco y realiza un cierre formal del archivo a nivel del sistema operativo. | fclose(archivo) |
| fgets | char \*fgets(char \*buffer, int tamaño, FILE \*apArch); | pueden leer cadenas de un archivo | fgets (caracteres, num caracteres, archivo); |
| fprintf | int fprintf(FILE \*apArch, char \*formato, ...) | se comporta como un printf normal, pero imprime sobre un archivo txt | fprintf(archivo, escribir); fprintf(archivo, "%s", "\n"); |
| fscanf | int fscanf(FILE \*apArch, char \*formato, ...); | Se comporta con un scanf pero a nivel de archivo txt | fscanf(archivo, "%s", caracteres); |
| fread | int fread(void \*ap, size\_t tam, size\_t nelem, FILE \*archivo) | permite leer uno o varios elementos de la misma longitud a partir de la dirección de un apuntador | fread(tabla2, 4, 4, mihandle); |
| fwrite | int fwrite(void \*ap, size\_t tam, size\_t nelem, FILE \*archivo) | Escribe carácter(es) sobre un archivo de la misma longitud almacenados a partir de una dirección de memoria. El valor de retorno es el número de elementos escritos. | fwrite(buffer, 1, bytesLeidos, archivoSalida); |