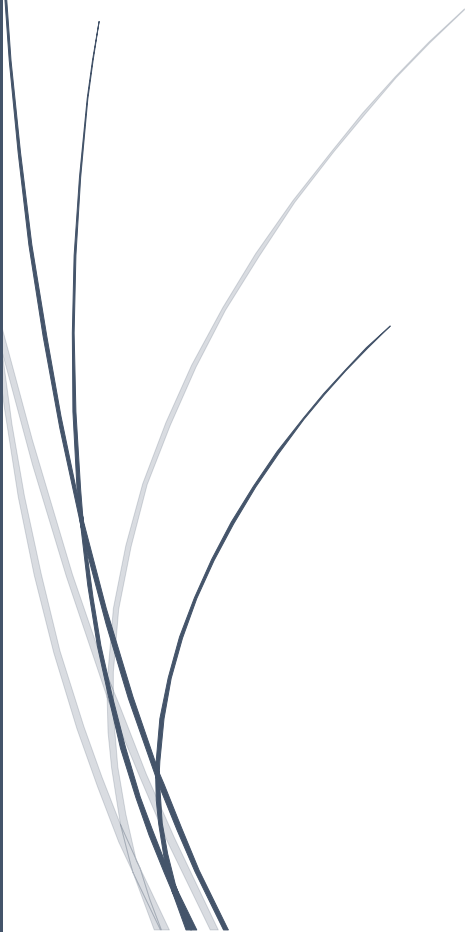


A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

10/20/2021

10 Ejercicios salteados

Práctica Tema 1

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Alejandro Colmenero Moreno
ASIGNATURA

ÍNDICE

EJERCICIO 1	3
Enunciado.....	3
Código	3
Resultados.....	3
EJERCICIO 2	5
Enunciado.....	5
Código	5
Resultados.....	6
EJERCICIO 3	7
Enunciado.....	7
Código	7
Resultados.....	8
EJERCICIO 4	10
Enunciado.....	10
Código	10
Resultados.....	11
EJERCICIO 5	12
Enunciado.....	12
Código	12
Resultados.....	13
EJERCICIO 6	15
Enunciado.....	15
Código	15
Resultados.....	17
EJERCICIO 7	18
Enunciado.....	18
Código	18
Resultados.....	20
EJERCICIO 8	21
Enunciado.....	21
Código	21
Resultados.....	22
EJERCICIO 9	23
Enunciado.....	23
Código	23
Resultados.....	25
EJERCICIO 10	26
Enunciado.....	26

Código	26
Resultados.....	32
BIBLIOGRAFÍA.....	37

EJERCICIO 1

Enunciado

1. Guardar Cadena - Realizar un programa en C que lea una cadena por teclado y la guarde en el fichero de texto "fich1.txt".

Código

```
#include <stdio.h>

#define MAX_TAM 11

int main()
{
    char string[MAX_TAM];
    FILE *file = fopen("fich1.txt", "w");

    if(file == NULL){

        printf("Hubo un error al abrir el archivo");

    } else {

        printf("Escribe una cadena de %i o menos caracteres:\n", MAX_TAM-1);

        // fGet porque sino da error en caso que el usuario se pase de caracteres
        fgets(string, MAX_TAM, stdin);

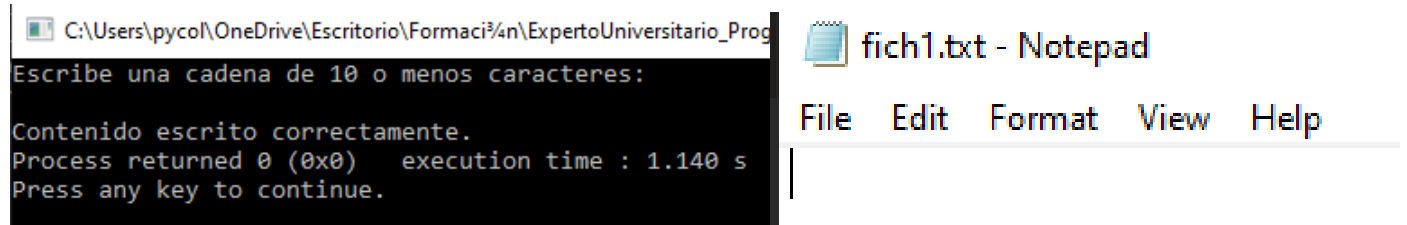
        fputs(string, file);

    }
    fclose(file);

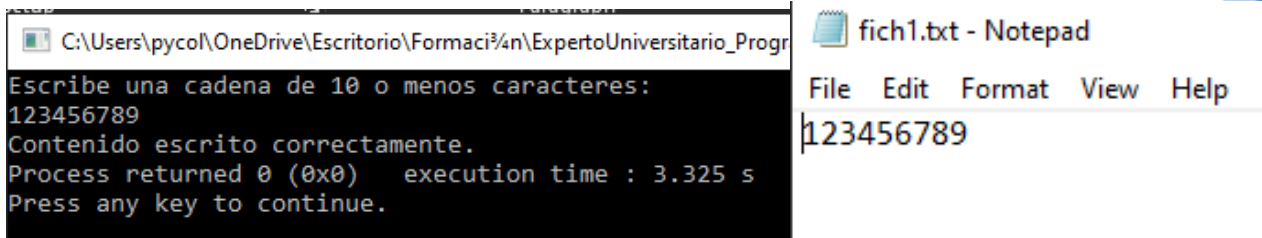
    return 0;
}
```

Resultados

No introducir nada



Introducir menos de lo máximo



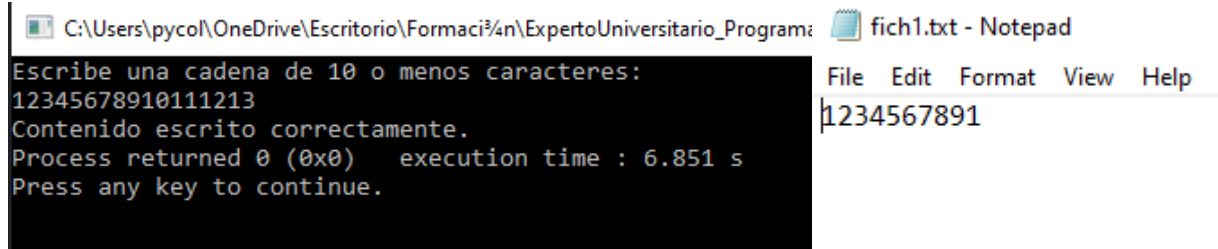
```
C:\Users\pycol\OneDrive\Escritorio\Formaci34n\ExpertoUniversitario_Progr  
Escribe una cadena de 10 o menos caracteres:  
123456789  
Contenido escrito correctamente.  
Process returned 0 (0x0) execution time : 3.325 s  
Press any key to continue.
```

fich1.txt - Notepad

File Edit Format View Help

123456789

Introducir más del máximo.



```
C:\Users\pycol\OneDrive\Escritorio\Formaci34n\ExpertoUniversitario_Program:  
Escribe una cadena de 10 o menos caracteres:  
12345678910111213  
Contenido escrito correctamente.  
Process returned 0 (0x0) execution time : 6.851 s  
Press any key to continue.
```

fich1.txt - Notepad

File Edit Format View Help

1234567891

EJERCICIO 2

Enunciado

4. Asterisco - Realizar un programa en C que lea caracteres por teclado hasta escribir un '*'. Guardar todos los caracteres pulsados en el fichero "fich2.txt".

Código

```
#include <stdio.h>

#define MAX_TAM 11

int main()
{
    char string[MAX_TAM];
    // cuando imprimo por pantalla el string, los espacios
    // no asignados los rellena con caracteres basura
    for (int x = 0; x < MAX_TAM-1; x++) string[x] = ' ';

    // instanciar el archivo
    FILE *file = fopen("fich2.txt", "w");

    if(file == NULL){

        printf("Hubo un error al abrir el archivo");

    } else {

        printf("Escribe una caracteres de uno en uno, máximo %i:\n", MAX_TAM-1);

        // fGet porque sino da error en caso que el
        // usuario se pase de caracteres
        int index = 0;
        char current_char;

        do{
            // para borrar el buffer porque pilla el ENTER
            // de la anterior vez y sigue para adelante sin preguntar
            fseek(stdin,0,SEEK_END);
            // carácter introducido por comando
            scanf("%c", &current_char);
            // no meter el * en el string
            if(current_char!='*') {
                // agregar el CHAR al CHAR[] si no es un salto de linea
                if(current_char!='\n') string[index] = current_char;
                index++;
            }
        }
        while( (index<MAX_TAM-1) && (current_char!='*') );

        // introducimos el CHAR[] en el FILE
        fputs(string, file);

        printf("Contenido escrito correctamente: %s\n", string);
```

```

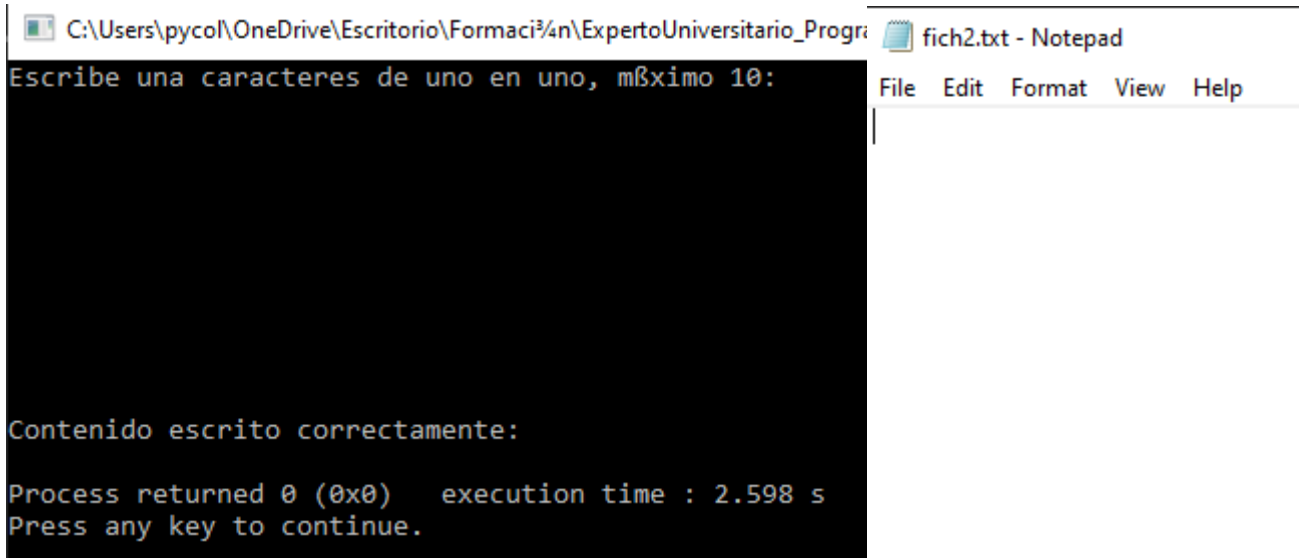
}
// cierro el archivo
fclose(file);

return 0;
}

```

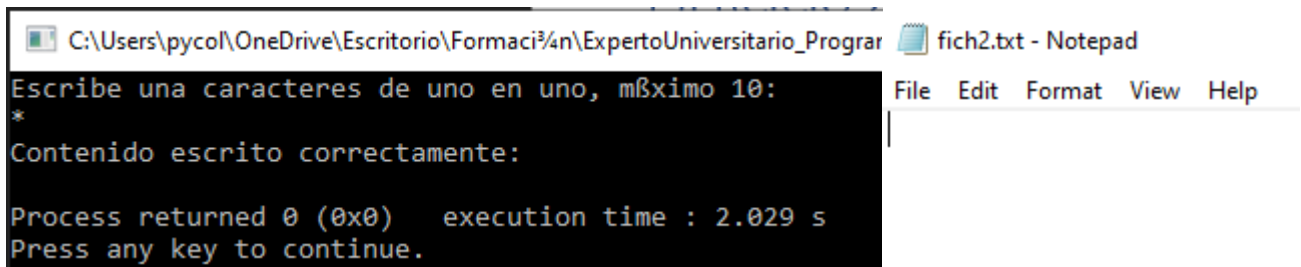
Resultados

Todos saltos de línea:



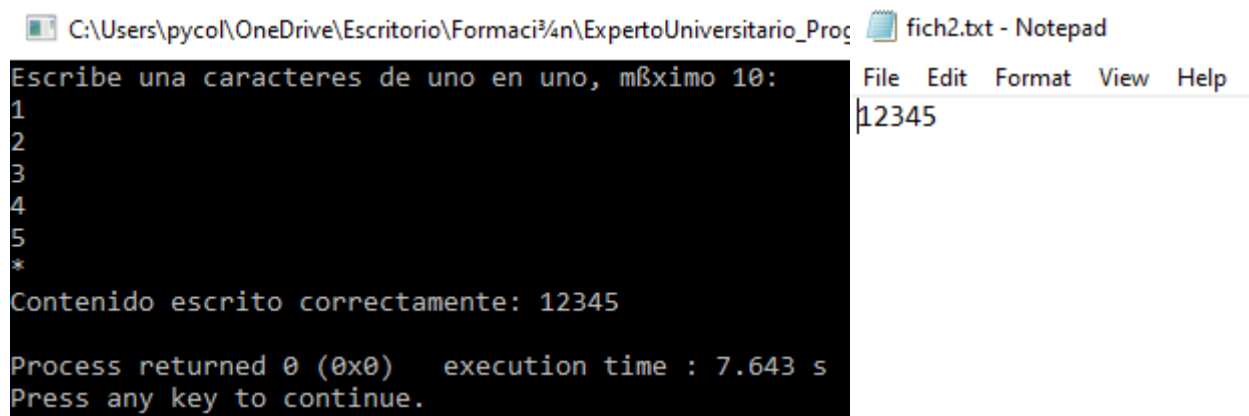
The screenshot shows a Windows command prompt window titled "C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Progra" and a Notepad window titled "fich2.txt - Notepad". The command prompt displays the prompt "Escribe una caracteres de uno en uno, m3ximo 10:" and the output "Contenido escrito correctamente:". Below this, it shows "Process returned 0 (0x0) execution time : 2.598 s" and "Press any key to continue.".

Introducir * desde el principio:



The screenshot shows a Windows command prompt window titled "C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Progra" and a Notepad window titled "fich2.txt - Notepad". The command prompt displays the prompt "Escribe una caracteres de uno en uno, m3ximo 10:" and the output "Contenido escrito correctamente:". Below this, it shows "Process returned 0 (0x0) execution time : 2.029 s" and "Press any key to continue.".

Introducir caracteres y parar a la mitad:



The screenshot shows a Windows command prompt window titled "C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Progra" and a Notepad window titled "fich2.txt - Notepad". The command prompt displays the prompt "Escribe una caracteres de uno en uno, m3ximo 10:" and the output "Contenido escrito correctamente: 12345". Below this, it shows "Process returned 0 (0x0) execution time : 7.643 s" and "Press any key to continue.".

EJERCICIO 3

Enunciado

8. Contar - Realizar un programa en C que lea el nombre de un fichero de texto por teclado y nos cuente el número de vocales, el número de consonantes y el número de signos de puntuación (incluido el espacio) que tiene. Mostrar por pantalla el resultado.

Código

```
#include <stdio.h>

#define VOWELS "aeiou"
#define CONSONANTS "bcdfghjklmnpqrstvwxyz"
#define NUMBERS "0123456789"
#define MAX_TAM 2

int main()
{

    char filename[50];
    FILE *file;

    int vowels = 0;
    int consonants = 0;
    int numbers = 0;
    int signs = 0;

    // get filename and file
    printf("Escribe el nombre del archivo: ");
    gets(filename);
    file = fopen(filename, "r");

    if(file == NULL){

        printf("Hubo un error al abrir el archivo\n");

    } else { // si existe el archivo

        // recorremos el archivo de CHAR en CHAR
        char string[2];
        while(fgets(string, MAX_TAM, file) != NULL){

            // cojo el CHAR y lo pongo en minúsculas
            char current_char = string[0];
            current_char = tolower(current_char);

            // si el carácter no es NULO
            if( current_char != 0 ){

                // comprobamos para cada uno de los grupos de caracteres
                int isvowel = strchr(VOWELS, current_char);
                int isconsonant = strchr(CONSONANTS, current_char);
                int isnumber = strchr(NUMBERS, current_char);
```



```

        // comprobamos cual es el afortunado
        if(isvowel != 0) {
            vowels++;
        } else if(isconsonant != 0) {
            consonants++;
        } else if(isnumber != 0) {
            numbers++;
        } else if(current_char != "\n") { // si no es ninguno, pero tampoco un
salto de linea.
            signs++;
        }
    }
}

// imprimir los datos
printf("\nDatos del archivo %s:", filename);
printf("\nVOCALES: %i", vowels);
printf("\nCONSONANTES: %i", consonants);
printf("\nNUMEROS: %i", numbers);
printf("\nSIGNOS: %i\n", signs);

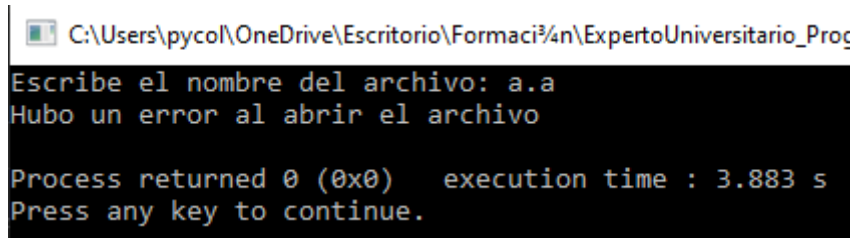
}
// cerrar archivo
fclose(file);

return 0;
}

```

Resultados

Error al introducir un nombre de fichero no existente:



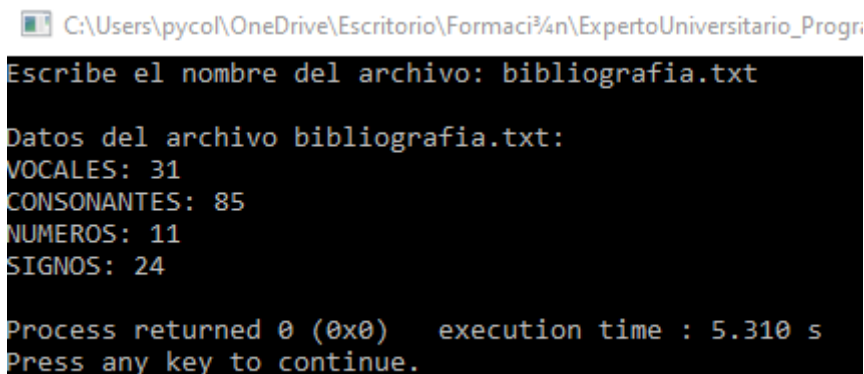
```

C:\Users\pycol\OneDrive\Escritorio\Formaci3\4n\ExpertoUniversitario_Progr
Escribe el nombre del archivo: a.a
Hubo un error al abrir el archivo

Process returned 0 (0x0)   execution time : 3.883 s
Press any key to continue.

```

Archivo bien escrito 1:



```


C:\Users\pycol\OneDrive\Escritorio\Formaci3\4n\ExpertoUniversitario_Progr
Escribe el nombre del archivo: bibliografia.txt

Datos del archivo bibliografia.txt:
VOCALES: 31
CONSONANTES: 85
NUMEROS: 11
SIGNOS: 24

Process returned 0 (0x0)   execution time : 5.310 s
Press any key to continue.

```

Archivo bien escrito 2:

 C:\Users\pycol\OneDrive\Escritorio\Formaci³4n\ExpertoUniversitario_Prog

Escribe el nombre del archivo: fich1.txt

Datos del archivo fich1.txt:

VOCALES: 1


CONSONANTES: 1

NUMEROS: 10

SIGNOS: 1

Process returned 0 (0x0) execution time : 1.081 s

Press any key to continue.

 fich1.txt - Notepad

File Edit Format View Help

1234fe-567891

EJERCICIO 4

Enunciado

12. Entero - Realizar un programa en C que guarde en el fichero binario "uno.bin" un entero dado por teclado.

Código

```
#include <stdio.h>

int main()
{
    int numero = 0;
    FILE *file1, *file2;
    char filename[8] = "uno.bin";
    file1 = fopen(filename, "wb");

    if(file1 == NULL){

        printf("Hubo un error al abrir el archivo");

    } else {

        // ESCRIBIR EN EL ARCHIVO
        // escanea el INT

        printf("Escribe un numero entero: ");
        scanf("%d", &numero);
        // escribimos en binario el número
        fwrite(&numero, sizeof numero, 1, file1);
        fclose(file1);

        // LEER EL INTERIOR DEL ARCHIVO
        numero = 0; // reinicio la variable
        // abro el archivo de nuevo, pero esta vez indico que es lectura
        file2 = fopen(filename, "rb");
        if(file2 == NULL){
            printf("Hubo un error al abrir el archivo");
        } else {
            // lo leo y lo meto en la variable
            fread(&numero, sizeof numero, 1, file2);
            printf("\nContenido: %i \n", numero);
            fclose(file2);
        }
    }

    return 0;
}
```

Resultados

Introducir un valor no numérico:

```
C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Progra
Escribe un numero entero: g
Contenido: 0
Process returned 0 (0x0)   execution time : 1.820 s
Press any key to continue.
```

Introducir un valor numérico:

```
C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Progra
Escribe un numero entero: 777
Contenido: 777
Process returned 0 (0x0)   execution time : 1.956 s
Press any key to continue.
```

Introducir un valor numérico y no numérico al final:

```
C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Progra
Escribe un numero entero: 3675ç
Contenido: 3675
Process returned 0 (0x0)   execution time : 3.487 s
Press any key to continue.
```

EJERCICIO 5

Enunciado

17. Tabla 10 reales - Realizar un programa en C que lea una tabla de 10 números reales y los guarde en un fichero binario llamado "tabla3.dat"

Código

```
#include <stdio.h>

#define TAM 6

int main()
{
    FILE *file1, *file2;
    float current_number = 0;
    float list[TAM];

    file1 = fopen("tabla3.dat", "wb");

    if(file1 == NULL){
        printf("Hubo un error al abrir el archivo");
    } else {

        // ESCRIBIR
        // Leemos por consola los números
        printf("Escribe %i numeros:\n", TAM);
        for(int x = 0; x < TAM; x++){
            // limpio el buffer, pq si el usuario introduce una letra
            // este dejará de escanear
            fseek(stdin,0,SEEK_END);
            // pido el float
            printf("Numero %02i: ", x+1);
            scanf("%f", &current_number);
            // guardo el valor en la lista
            list[x] = current_number;
        }
        // escribirmos la lista como binario
        fwrite(&list, sizeof list, 1, file1);
        fclose(file1);

        // LEER
        if(file2 == NULL){
            printf("Hubo un error al abrir el archivo");
        } else {

            file2 = fopen("tabla3.dat", "rb");
            float buffer[TAM];

            // hago cast a (void*) porque la funcion FREAD no acepta
            // como primer parametro una variable de tipo FLOAT
            fread((void*)&buffer, sizeof(buffer), 1, file2);
            // itramos el resultado y lo imprimimos por pantalla
```

```

        for(int x = 0; x < TAM; x++){
            printf("\n%f", buffer[x]);
        }
        fclose(file2);
    }
}

return 0;
}

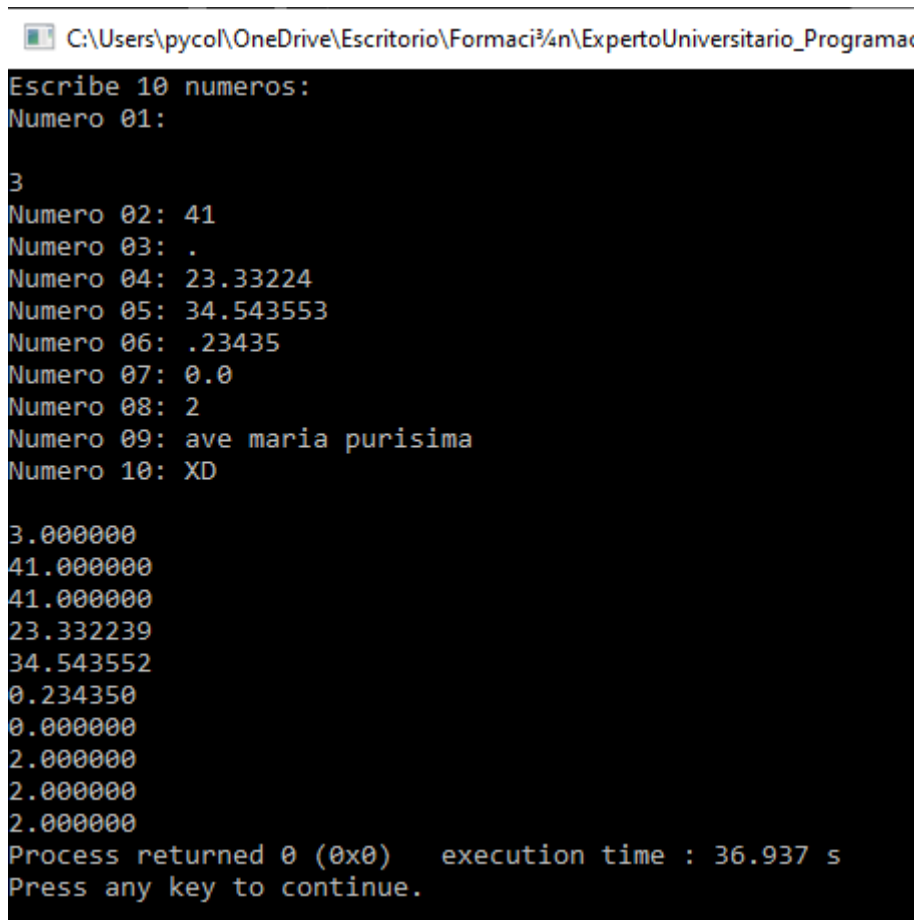
fclose(file);
}

return 0;
}

```

Resultados

Intentando poner saltos de línea y caracteres que no son números:



```

C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Programac
Escribe 10 numeros:
Numero 01:
3
Numero 02: 41
Numero 03: .
Numero 04: 23.33224
Numero 05: 34.543553
Numero 06: .23435
Numero 07: 0.0
Numero 08: 2
Numero 09: ave maria purisima
Numero 10: XD

3.000000
41.000000
41.000000
23.332239
34.543552
0.234350
0.000000
2.000000
2.000000
2.000000
Process returned 0 (0x0) execution time : 36.937 s
Press any key to continue.

```

En el primer número introducido, cuando le das a ENTER, no avanza hasta que pongas cualquier otra cosa.

Introduciendo números:

[illegible]

Deduzco por lo que veo en la imagen, que el redondeo en números de punto flotante no es muy exacto.

EJERCICIO 6

Enunciado

21. Notas - Realizar un programa en C que dado el tipo Notas, con un código, un nombre, un curso y una nota, lea por teclado una tabla de 5 elementos del tipo Notas. El programa guardará dicha tabla en un fichero binario llamado "notas.bin".

Código

```
#include <stdio.h>

#define TAM 5

typedef struct NOTA{
    int codigoNota;
    char nombreAlumno[50];
    char cursoAlumno[5];
    float nota;
};

int main()
{
    FILE *fileW, *fileR;
    struct NOTA notasW[TAM];

    fileW = fopen("notas.bin", "wb");
    if(fileW == NULL){
        printf("Hubo un error al abrir el archivo");
    } else {

        // ESCRIBIR
        printf("Escribe %i notas:\n", TAM);
        // pedimos cada uno de los atributos por consola
        for(int x = 0; x < TAM; x++){
            //fseek(stdin,0,SEEK_END);
            fseek(stdin,0,SEEK_END);
            printf("\nNOTA %i%c\n", x+1,-89);
            // código aleatorio
            notasW[x].nota = rand();

            // Otros atributos por consola
            printf("Nombre: ");
            // fgets para que no supere lo máximo
            fgets((void*)&notasW[x].nombreAlumno,sizeof(notasW[x].nombreAlumno), stdin);
            // borramos el \n que se queda al final de la variable
            strtok(&notasW[x].nombreAlumno, "\n");
            // borramos el buffer para que no se quede el \n al final
            fseek(stdin,0,SEEK_END);

            printf("Curso: ");
            // fgets para que no supere lo máximo
            fgets((void*)&notasW[x].cursoAlumno,sizeof(notasW[x].cursoAlumno), stdin);
```



```

    // borramos el \n que se queda al final de la variable
    strtok(&notasW[x].cursoAlumno, "\n");
    // borramos el buffer para que no se quede el \n al final
    fseek(stdin,0,SEEK_END);

    printf("Nota: ");
    scanf("%f", &notasW[x].nota);
}

// escribir las notas en binario
fwrite(notasW, sizeof(notasW), 1, fileW);
fclose(fileW);

// LEER
printf("\nLEYENDO BINARIO");
struct NOTA notasR[TAM];
fileR = fopen("notas.bin", "rb");
if(fileR == NULL){
    printf("Hubo un error al abrir el archivo");
} else {
    fread(&notasR,sizeof(notasR),1,fileR);
    for(int x = 0; x < TAM; x++){
        printf("\nNOTA %i%c\n", x+1,-89);
        printf("Código: %i\n", notasR[x].codigoNota);
        printf("Nombre: %s\n", notasR[x].nombreAlumno);
        printf("Curso: %s\n", notasR[x].cursoAlumno);
        printf("Nota: %f\n", notasR[x].nota);
    }
    close(fileR);
}
}

return 0;
}

```

Resultados

1. Poniendolo bien.
2. Intentando dejarlo vacío.
3. Superando los límites.

[illegible]

EJERCICIO 7

Enunciado

24. Ordenar Notas - Realizar un programa en C que lea el fichero binario "notas.bin", ordene dicho fichero por nota y lo guarde en otro fichero binario "notas.ord".

Código

```
#include <stdio.h>

#define TAM 5

typedef struct NOTA{
    int codigoNota;
    char nombreAlumno[50];
    char cursoAlumno[3];
    float nota;
};

int main()
{
    FILE *fileW, *fileR;
    struct NOTA notas[TAM];
    struct NOTA current_nota;

    fileR = fopen("notas.bin", "rb");
    if(fileR == NULL){
        printf("Hubo un error al abrir el archivo");
    } else {

        // LEER
        fread(&notas,sizeof(notas),1,fileR);

        // mostrar antes de ordenar
        for(int x = 0; x < TAM; x++){
            printf("\nNOTA %i%c\n", x+1,-89);
            printf("Código: %i\n", notas[x].codigoNota);
            printf("Nombre: %s\n", notas[x].nombreAlumno);
            printf("Curso: %s\n", notas[x].cursoAlumno);
            printf("Nota: %f\n", notas[x].nota);
        }

        // tipical metetodo de ordenación no copiado de google
        for (int x = 0; x < TAM; x++){
            for (int y = x + 1; y <TAM; y++){
                if (notas[x].nota < notas[y].nota){
                    current_nota = notas[x];
                    notas[x] = notas[y];
                    notas[y] = current_nota;
                }
            }
        }
    }
}
```

```
// mostrar despues de ordenar
for(int x = 0; x < TAM; x++){
    printf("\nNOTA %i%c\n", x+1,-89);
    printf("Código: %i\n", notas[x].codigoNota);
    printf("Nombre: %s\n", notas[x].nombreAlumno);
    printf("Curso: %s\n", notas[x].cursoAlumno);
    printf("Nota: %f\n", notas[x].nota);
}
fclose(fileR);

// ESCRIBIR
fileW = open("notas.ord","wb");
if(fileW!=NULL){
    printf("Hubo un error al abrir el archivo");
} else {
    fwrite(notas, sizeof(notas), 1, fileW);
    fclose(fileW);
    printf("Escrito correctamente y ordenado");
}

}

return 0;
}
```

Resultados

```
C:\Users\pycol\OneDrive\Escritorio\Formaci%4n\ExpertoUniversitario_Programacion.  
NOTA 1ª  
C% digo: 7471307  
Nombre: Alejandro  
Curso: 3f  
Nota: 4.000000  
  
NOTA 2ª  
C% digo: 6422044  
Nombre: Pedro  
Curso: 2g  
Nota: 8.000000  
  
NOTA 3ª  
C% digo: 0  
Nombre: Maria  
Curso: 6a  
Nota: 8.000000  
  
NOTA 4ª  
C% digo: 6422112  
Nombre: Marta  
Curso: 6b  
Nota: 1.000000  
  
NOTA 5ª  
C% digo: 6422124  
Nombre: Laura  
Curso: 9e  
Nota: 10.000000  
  
NOTA 1ª  
C% digo: 6422124  
Nombre: Laura  
Curso: 9e  
Nota: 10.000000  
  
NOTA 2ª  
C% digo: 0  
Nombre: Maria  
Curso: 6a  
Nota: 8.000000  
  
NOTA 3ª  
C% digo: 6422044  
Nombre: Pedro  
Curso: 2g  
Nota: 8.000000  
  
NOTA 4ª  
C% digo: 7471307  
Nombre: Alejandro  
Curso: 3f  
Nota: 4.000000  
  
NOTA 5ª  
C% digo: 6422112  
Nombre: Marta  
Curso: 6b  
Nota: 1.000000  
Hubo un error al abrir el archivo  
Process returned 0 (0x0)   execution time : 0.053 s  
Press any key to continue.
```

EJERCICIO 8

Enunciado

27. Listín - Realizar un programa en C que dado el tipo Listín compuesta por un nombre, una dirección y un teléfono, genere un fichero binario con una tabla de 20 elementos de tipo Listín. En los campos nombre y apellidos meteremos "Vacio" y en el campo teléfono meteremos un 0.

Código

```
#include <stdio.h>

#define TAM 1

typedef struct LISTIN{
    char nombre[50];
    char direccion[100];
    int telefono;
};

int main()
{
    // GENERAR LISTINES VACIOS
    struct LISTIN listines[TAM];
    // 20 veces asignar un LISTIN "vacio"
    for(int x = 0; x < TAM; x++){

        // memcpy porque sino me da error al asignar implicitamente un array
        memcpy(listines[x].nombre, "", 1);
        memcpy(listines[x].direccion, "", 1);
        listines[x].telefono = 0;

    }
    printf("\nListines creados, pesan %i.\n", sizeof(listines));

    // ESCRIBIR ARCHIVO BINARIO;
    FILE *fileW;
    fileW = fopen("listines.bin","wb");

    if(fileW != NULL){
        // escribir
        fwrite(&listines, sizeof(listines), 1, fileW);
        fclose(fileW);
        printf("Archivo generado correctamente");
    } else {
        printf("Hubo un error al abrir el archivo");
    }

    return 0;
}
```

Resultados



Select C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_I

```
Listines creados, pesan 156.
Archivo generado correctamente
Process returned 0 (0x0)   execution time : 0.044 s
Press any key to continue.
```

Como veo tontería mostrar por pantalla el interior del archivo binario puesto que serían 20 ceros y 40 cadenas vacías, pues muestro su interior en hexadecimal con un programa, por si sirve de algo.

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII or .	
000000	00	FE	61	00	DC	2D	78	00	10	17	40	00	00	00	00	00	. . a . . - x .	. . @
000010	14	2C	78	00	60	FE	61	00	F9	6E	C0	76	14	2C	78	00	. , x . . a .	. n . v . , x .
000020	58	FE	61	00	95	6F	C1	76	1C	43	C6	76	98	FE	61	00	X . a . . o . v	. C . v . . a .
000030	F1	64	00	76	08	00	00	00	3D	6E	C0	76	23	6E	C0	76	. d . v	= n . v # n . v
000040	BC	A6	58	E4	DC	2D	78	00	88	00	00	00	00	00	00	00	. . X . . - x
000050	10	17	40	00	6C	FE	61	00	30	01	00	00	CC	FF	61	00	. . @ . l . a .	0 a .
000060	C0	CC	C0	76	EC	D7	FC	92	FE	FF	FF	FF	68	FF	61	00	. . . v h . a .
000070	CF	14	40	00	10	17	40	00	DE	5D	8E	77	00	00	00	00	. . @ . . @ .	.] . w
000080	01	00	00	00	88	00	00	00	DC	2D	78	00	68	FF	61	00 - x . h . a .
000090	7B	17	40	00	10	17	40	00	00	00	00	00					{ . @ . . @

EJERCICIO 9

Enunciado

30. Comparar Binarios - Realizar un programa en C que lea el nombre de dos ficheros binarios y compruebe si son iguales o no. Los ficheros deberán estar compuestos por una tabla de 5 enteros cada uno.

Código

```
#include <stdio.h>

#define TAM 1

typedef struct LISTIN{
    char nombre[50];
    char direccion[100];
    int telefono;
};

int get_file_length(FILE *file);

int main()
{
    // ESCRIBIR ARCHIVO BINARIO;
    FILE *fileW1, *fileW2;
    char filename01[50];
    char filename02[50];

    printf("Escribe el primer archivo: ");
    scanf("%s",filename01);
    printf("Escribe el segundo archivo: ");
    scanf("%s",filename02);

    fileW1 = fopen(filename01,"rb");
    fileW2 = fopen(filename02,"rb");

    // así consigo el tamaño del archivo
    int length1 = get_file_length(fileW1);
    int length2 = get_file_length(fileW2);

    // si los archivos no pesan lo mismo, directamente descartado
    if(length1 == length2){

        // le pongo el peso exacto del archivo
        char contenido01[length1];
        char contenido02[length2];

        if(fileW1 != NULL){

            if(fileW2 != NULL){

                // escribir
```



```

        fread(&contenido01,sizeof(contenido01),1,fileW1);
        fread(&contenido02,sizeof(contenido02),1,fileW2);

        if (strcmp(contenido01, contenido02) == 0) {
            printf("El archivo %s y el archivo %s son iguales.\n", filename01,
filename02);
        } else {
            printf("El archivo %s y el archivo %s NO son iguales.\n", filename01,
filename02);
        }

        } else {
            printf("Hubo un error al abrir el segundo archivo");
        }

        } else {
            printf("Hubo un error al abrir el primer archivo");
        }

        } else {
            printf("-El archivo %s y el archivo %s NO son iguales.\n", filename01,
filename02);
        }

        return 0;
    }

int get_file_length(FILE *file){

    fseek(file, 0, SEEK_END);
    // así consigo el tamaño del archivo
    int file_size = ftell(file);
    // vuelvo al inicio
    fseek(file, 0, SEEK_SET);
    return file_size;
}

```

Resultados

El archivo **a.bin** y **b.bin** contienen el INT 12 y el **c.bin** el INT 777.

Dos archivos iguales.

```
C:\Users\pycol\OneDrive\Escritorio\Formaci34n\ExpertoUniversitario_Programa
Escribe el primer archivo: a.bin
Escribe el segundo archivo: b.bin
El archivo a.bin y el archivo b.bin son iguales.

Process returned 0 (0x0)   execution time : 3.827 s
Press any key to continue.
```

	+0	+1	+2	+3	+4		+0	+1	+2	+3	+4	+5
000000	0C	00	00	00	00		000000	0C	00	00	00	

Los dos no son iguales.

```
C:\Users\pycol\OneDrive\Escritorio\Formaci34n\ExpertoUniversitario_Progr
Escribe el primer archivo: a.bin
Escribe el segundo archivo: c.bin
El archivo a.bin y el archivo c.bin NO son iguales.

Process returned 0 (0x0)   execution time : 19.246 s
Press any key to continue.
```

	+0	+1	+2	+3	+4		+0	+1	+2	+3	+4	+
000000	0C	00	00	00	00		000000	09	03	00	00	

EJERCICIO 10

Enunciado

35. Facturas - Realizar un programa en C que gestione la facturación de una empresa. Las facturas constarán de un campo código, una fecha y un importe. El programa permitirá crear una factura, borrar una factura, mostrar por pantalla todas las facturas y calcular la suma total de las facturas. Las facturas serán guardadas en un fichero binario llamado "Fact.bin" donde tendremos una tabla de 50 facturas.

Código

```
#include <stdio.h>
#include <stdlib.h>

typedef struct FACTURA{
    int codigo;
    int dia;
    int mes;
    int anyo;
    int importe;
};

int FACTURA_LENGTH;

int get_file_length(FILE *file);
int get_factura_size();

void show_menu();
void delete_factura();
void show_facturas();
void addup_factura();

const char FILENAME[] = "facturas.bin";

int main(){

    FACTURA_LENGTH = get_factura_struct_size();

    show_menu();

    return 0;
}

void create_factura(){

    FILE *fileW;
    struct FACTURA current_factura;
    fileW = fopen(FILENAME, "ab");

    if(fileW == NULL){
        printf("Hubo un error al abrir el archivo");
    } else {
```

```

printf("ESCRIBE LA FACTURA:\n");

// Otros atributos por consola
// CODIGO
int codigo = time(0);

// FECHA
int dia = 0, mes = 0, anyo = 2021;
printf("Fecha:\n");
printf("  Día: ");
scanf("%i", &dia);
printf("  Mes: ");
scanf("%i", &mes);
printf(" Anyo: ");
scanf("%i", &anyo);

// IMPORTE
int importe = 0;
printf("Importe: ");
scanf("%i", &importe);

// ESCRIBIR LOS DATOS
current_factura.codigo = codigo;
current_factura.dia = dia;
current_factura.mes = mes;
current_factura.anyo = anyo;
current_factura.importe = importe;

// GUARDAR Y CERRAR
fwrite(&current_factura, sizeof(current_factura), 1, fileW);
fclose(fileW);

// TERMINAR
system("cls");
printf("\nDatos escritos correctamente.\n\n");
show_menu();
}
}

void delete_factura(){

// LEAMOS Y GUARDAMOS TODAS LAS FACTURAS ANTES DE ESCRIBIR Y PERDERLAS
FILE *fileR;
fileR = fopen(FILENAME, "rb");

int file_length = 0;
int facturas_ammount = 0;

if(fileR == NULL){
    printf("Hubo un error al abrir el archivo");
} else {

    // get informacion del archivo
    file_length = get_file_length(fileR);

```

```

facturas_ammount = file_length/FACTURA_LENGTH;

// GUARDAMOS LAS FACTURAS
struct FACTURA last_facturas[facturas_ammount];
fread(&last_facturas,sizeof(last_facturas),1,fileR);

// ESTRUCTURA SIN LA FACTURA CORRESPONDIENTE
struct FACTURA new_facturas[facturas_ammount-1];

int factura_index = 0;
printf("\nQue factura quieres borrar? ");
for(int x = 0; x < facturas_ammount; x++){
    printf("\nFACTURA %i: ", x+1);
    printf("Codigo: %i, ", last_facturas[x].codigo);
    printf("Fecha: %02i/%02i/%04i, ", last_facturas[x].dia, last_facturas[x].mes,
last_facturas[x].anyo);
    printf("Importe: %i\\$", last_facturas[x].importe);
}
printf("\n: ");
scanf("%i", &factura_index);
if(factura_index > 0 && factura_index <= facturas_ammount){ // si la factura
existe

    char confirm = ' ';
    printf("\nSeguro que quieres borrar esta factura?(Y/N)");
    do{
        // borramos el buffer para que no se quede el \n al final
        fseek(stdin,0,SEEK_END);
        printf(": ");
        scanf("%c", &confirm);
        if(confirm == 'Y' || confirm == 'y'){
            // borrar

            // CREAR NUEVA LISTA DE FACTURAS SIN LA SELECCIONADA
            int index = 0;
            for(int x = 0; x < facturas_ammount; x++){
                if((factura_index-1) != x){
                    new_facturas[index] = last_facturas[x];
                    index++;
                }
            }

            // ESCRIBIMOS EN EL ARCHIVO BINARIO
            FILE *fileW;
            fileW = fopen(FILENAME, "wb");

            if(fileW != NULL){
                fwrite(new_facturas, sizeof new_facturas, 1, fileW);
                fclose(fileW);

                system("cls");
                printf("\nFactura borrada correctamente.\n");
                show_menu();
            }
        }
    } while(confirm != 'Y' && confirm != 'y');
}
}

```

```

        } else {
            printf("Hubo un error al abrir el archivo");
        }

        } else if(confirm == 'N' || confirm == 'n'){
            system("cls");
            printf("\nFactura intancata.\n");
            show_menu();
        } else {
            printf("\nNo te he entendido, escribe Y o N");
        }
    } while(confirm != 'Y' && confirm != 'N' && confirm != 'y' && confirm != 'n');

} else {
    printf("\nFactura no encontrada.\n");
    show_menu();
}
}

void show_facturas(){

    FILE *fileR;
    fileR = fopen(FILENAME, "rb");

    int file_length = 0;
    int facturas_ammount = 0;

    if(fileR == NULL){
        printf("Hubo un error al abrir el archivo");
    } else {

        // get informacion del archivo
        file_length = get_file_length(fileR);
        facturas_ammount = file_length/FACTURA_LENGTH;

        // crear STRUCT y guardar datos
        struct FACTURA facturas[facturas_ammount];
        fread(&facturas,sizeof(facturas),1,fileR);

        // sacar por pantalla los datos
        printf("\nHay %i facturas guardadas: \n", facturas_ammount);
        for(int x = 0; x < facturas_ammount; x++){
            printf("\nFACTURA %i:
=====
Codigo: %i\n", facturas[x].codigo);
            printf(" Fecha: %02i/%02i/%04i\n", facturas[x].dia, facturas[x].mes,
facturas[x].anyo);
            printf(" Importe: %i$ \n", facturas[x].importe);
            printf("=====
");
        }
    }
}

```

```

    }
    printf("\n");
    show_menu();
}

void addup_factura(){

    FILE *fileR;
    fileR = fopen(FILENAME, "rb");

    int file_length = 0;
    int facturas_ammount = 0;

    int suma = 0;

    if(fileR!=NULL){

        // get informacion del archivo
        file_length = get_file_length(fileR);
        facturas_ammount = file_length/FACTURA_LENGTH;

        // GUARDAMOS LAS FACTURAS
        struct FACTURA facturas[facturas_ammount];
        fread(&facturas,sizeof(facturas),1,fileR);

        for(int x = 0; x < facturas_ammount; x++) {
            suma += facturas[x].importe;
            printf("FACTURA %i:\t%i$\t%i$\n", x,facturas[x].importe,suma);
        }
        printf("-----\n");
        printf("TOTAL: \t\t\t%i$\n\n",suma);

        show_menu();

    } else {
        printf("Hubo un error al abrir el archivo");
    }
}

void show_menu(){

    int index = 0;

    // borramos el buffer para que no haga bucle infinito
    fseek(stdin,0,SEEK_END);

    //printf("Elije qué quieres hacer: \n");
    printf("\nEscribe 1 si quieres MOSTRAR UNA FACTURA.\n");
    printf("Escribe 2 si quieres CREAR UNA FACTURA.\n");
    printf("Escribe 3 si quieres BORRAR FACTURAS.\n");
    printf("Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.\n");
    printf("Escribe 5 si quieres SALIR DEL PROGRAMA.\n");
    printf(": ");
}

```

```

scanf("%i", &index);

switch(index){
    case 2:
        system("cls");
        create_factura();
        break;
    case 3:
        system("cls");
        delete_factura();
        break;
    case 1:
        system("cls");
        show_facturas();
        break;
    case 4:
        system("cls");
        addup_factura();
        break;
    case 5:
        printf("Bye :)");
        break;
    default:

        system("cls");

        printf("\nNo te he entendido.\n\n");
        show_menu();
}

}

int get_file_length(FILE *file){

    fseek(file, 0, SEEK_END);
    // así consigo el tamaño del archivo
    int file_size = ftell(file);
    // vuelvo al inicio
    fseek(file, 0, SEEK_SET);
    return file_size;
}

int get_factura_struct_size(){

    struct FACTURA factura;
    return sizeof factura;
}

```


Resultados

El programa empieza enseñándote un menú donde puedes elegir que hacer.

```
C:\Users\pycol\OneDrive\Escritorio\Formaci³4n\ExpertoUniversitario_Prog
Escribe 1 si quieres MOSTRAR UNA FACTURA.
Escribe 2 si quieres CREAR UNA FACTURA.
Escribe 3 si quieres BORRAR FACTURAS.
Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.
Escribe 5 si quieres SALIR DEL PROGRAMA.
:
```

Pulsando 1, nos muestra todas las facturas registradas:

```
C:\Users\pycol\OneDrive\Escritorio\Formaci³4n\ExpertoUniversitario_ProgramacionAvanzada\FundamentosPi
Hay 3 facturas guardadas:

FACTURA 1: =====
Codigo: 233546
Fecha: 23/11/2021
Importe: 100$
=====

FACTURA 2: =====
Codigo: 1634829484
Fecha: 29/12/2020
Importe: 99$
=====

FACTURA 3: =====
Codigo: 1634831912
Fecha: 05/12/2017
Importe: 550$
=====

Escribe 1 si quieres MOSTRAR UNA FACTURA.
Escribe 2 si quieres CREAR UNA FACTURA.
Escribe 3 si quieres BORRAR FACTURAS.
Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.
Escribe 5 si quieres SALIR DEL PROGRAMA.
:
```

Pulsando 2 te pide los datos para crear una factura:

```
C:\Users\pycol\OneDrive\Escritoric
ESCRIBE LA FACTURA:
Fecha:
Dia: 13
Mes: 6
Anyo: 1999
Importe: 80
```

```

C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_Programac
Datos escritos correctamente.

Escribe 1 si quieres MOSTRAR UNA FACTURA.
Escribe 2 si quieres CREAR UNA FACTURA.
Escribe 3 si quieres BORRAR FACTURAS.
Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.
Escribe 5 si quieres SALIR DEL PROGRAMA.
:

```

Si vuelvo a escribir 1 vemos que hay una nueva factura al final. La nº 4, de 550\$:

```

C:\Users\pycol\OneDrive\Escritorio\Formaci3/4n\ExpertoUniversitario_ProgramacionAvanzada\FundamentosPr
Hay 4 facturas guardadas:

FACTURA 1: =====
Codigo: 233546
Fecha: 23/11/2021
Importe: 100$
=====

FACTURA 2: =====
Codigo: 1634829484
Fecha: 29/12/2020
Importe: 99$
=====

FACTURA 3: =====
Codigo: 1634831912
Fecha: 05/12/2017
Importe: 550$
=====

FACTURA 4: =====
Codigo: 1634832252
Fecha: 13/06/1999
Importe: 80$
=====

Escribe 1 si quieres MOSTRAR UNA FACTURA.
Escribe 2 si quieres CREAR UNA FACTURA.
Escribe 3 si quieres BORRAR FACTURAS.
Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.
Escribe 5 si quieres SALIR DEL PROGRAMA.
:

```

Si escribes 3, te da a elegir que factura quieres borrar. Te las enseña todas de manera más comprimida:

C:\Users\pycol\OneDrive\Escritorio\Formaci³4n\ExpertoUniversitario_ProgramacionAvanzada\FundamentosPro

```
Que factura quieres borrar?
FACTURA 1:Codigo: 233546, Fecha: 23/11/2021, Importe: 100$
FACTURA 2:Codigo: 1634829484, Fecha: 29/12/2020, Importe: 99$
FACTURA 3:Codigo: 1634831912, Fecha: 05/12/2017, Importe: 550$
FACTURA 4:Codigo: 1634832252, Fecha: 13/06/1999, Importe: 80$
:
```

Una vez escrita que factura quieres borrar, te pregunta si estás seguro. Si le dices que no:

C:\Users\pycol\OneDrive\Escritorio\Formaci³4n\ExpertoUniversitario_ProgramacionAvanzada\Funda

```
Que factura quieres borrar?
FACTURA 1:Codigo: 233546, Fecha: 23/11/2021, Importe: 100$
FACTURA 2:Codigo: 1634829484, Fecha: 29/12/2020, Importe: 99$
FACTURA 3:Codigo: 1634831912, Fecha: 05/12/2017, Importe: 550$
FACTURA 4:Codigo: 1634832252, Fecha: 13/06/1999, Importe: 80$
: 2
```

```
Seguro que quieres borrar esta factura?(Y/N): n
```

C:\Users\pycol\OneDrive\Escritorio\Formaci³4n\ExpertoUniversitario_Progr

```
Factura intancata.

Escribe 1 si quieres MOSTRAR UNA FACTURA.
Escribe 2 si quieres CREAR UNA FACTURA.
Escribe 3 si quieres BORRAR FACTURAS.
Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.
Escribe 5 si quieres SALIR DEL PROGRAMA.
:
```

Si le dices que sí:

C:\Users\pycol\OneDrive\Escritorio\Formaci³4n\ExpertoUniversitario_Program

```
Factura borrada correctamente.

Escribe 1 si quieres MOSTRAR UNA FACTURA.
Escribe 2 si quieres CREAR UNA FACTURA.
Escribe 3 si quieres BORRAR FACTURAS.
Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.
Escribe 5 si quieres SALIR DEL PROGRAMA.
:
```

Y si muestro las facturas, desaparece la que tenía de importe 99\$:

```
C:\Users\pycol\OneDrive\Escritorio\Formaci3\4n\ExpertoUniversitario_ProgramacionAvanzada\Fundamentos

Hay 3 facturas guardadas:

FACTURA 1: =====
Codigo: 233546
Fecha: 23/11/2021
Importe: 100$
=====

FACTURA 2: =====
Codigo: 1634831912
Fecha: 05/12/2017
Importe: 550$
=====

FACTURA 3: =====
Codigo: 1634832252
Fecha: 13/06/1999
Importe: 80$
=====

Escribe 1 si quieres MOSTRAR UNA FACTURA.
Escribe 2 si quieres CREAR UNA FACTURA.
Escribe 3 si quieres BORRAR FACTURAS.
Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.
Escribe 5 si quieres SALIR DEL PROGRAMA.
:
```

Por último, si pulsas 4 te muestra la suma del importe de todas las facturas:

```
C:\Users\pycol\OneDrive\Escritorio\Formaci3\4n\ExpertoUniversitario_ProgramacionAvanzada\

FACTURA 0:      100$      100$
FACTURA 1:      550$      650$
FACTURA 2:      80$       730$
-----
TOTAL:              730$

Escribe 1 si quieres MOSTRAR UNA FACTURA.
Escribe 2 si quieres CREAR UNA FACTURA.
Escribe 3 si quieres BORRAR FACTURAS.
Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.
Escribe 5 si quieres SALIR DEL PROGRAMA.
:
```

Al escribir 5, lo que hace es salir de la recursión de funciones por el que te lleva escribir uno u otro número, y por ende, se acaba la ejecución del programa:

```
C:\Users\pycol\OneDrive\Escritorio\Formaci3\4n\ExpertoUniversitario_ProgramacionAvan:

Escribe 1 si quieres MOSTRAR UNA FACTURA.
Escribe 2 si quieres CREAR UNA FACTURA.
Escribe 3 si quieres BORRAR FACTURAS.
Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.
Escribe 5 si quieres SALIR DEL PROGRAMA.
: 5
Bye :)

Process returned 0 (0x0)   execution time : 2.578 s
Press any key to continue.
```

Si escribes cualquier otra cosa:

C:\Users\pycol\OneDrive\Escritorio\Formaci34n\ExpertoUniversitario_Pro

No te he entendido.

Escribe 1 si quieres MOSTRAR UNA FACTURA.

Escribe 2 si quieres CREAR UNA FACTURA.

Escribe 3 si quieres BORRAR FACTURAS.

Escribe 4 si quieres CALCULAR SUMA DE FACTURAS.

Escribe 5 si quieres SALIR DEL PROGRAMA.

:

BIBLIOGRAFÍA

https://www.youtube.com/playlist?list=PLw8RQJQ8K1ySN6bVHYEpDoh-CKVkl_uOF

<https://stackoverflow.com/questions/9958000/how-does-scanf-handle-white-space>

<https://stackoverflow.com/questions/7880141/how-do-i-check-length-of-user-input-in-c>

<https://stackoverflow.com/questions/13542055/how-to-do-scanf-for-single-char-in-c>

<https://stackoverflow.com/questions/22676919/random-trailing-characters-printing-a-string-in-c>

<https://stackoverflow.com/questions/1422817/how-to-read-a-float-from-binary-file-in-c>

<https://www.javatpoint.com/array-of-structures-in-c>

<https://www.delftstack.com/howto/c/read-binary-file-in-c/>

<https://stackoverflow.com/questions/1472581/printing-chars-and-their-ascii-code-in-c>

<https://stackoverflow.com/questions/37225244/error-assignment-to-expression-with-array-type-error-when-i-assign-a-struct-f>

<https://stackoverflow.com/questions/36465942/error-when-trying-to-initialize-a-struct-in-c>

<https://fix.code-error.com/how-to-initialize-a-struct-in-accordance-with-c-programming-language-standards/>

<https://www.tutorialspoint.com/c-program-to-sort-an-array-in-an-ascending-order>

<https://www.w3schools.in/c-tutorial/function-arguments/>

<https://www.tutorialspoint.com/how-to-clear-console-in-c>

<https://easycodebook.com/2019/06/c-program-to-append-add-more-records-in-binary-file/>

<https://www.rapidtables.com/convert/number/hex-to-decimal.html>

<https://stackoverflow.com/questions/16623854/is-there-a-better-function-than-rand>

<https://h.markbuild.com/doc/binary-viewer.html>