Gonzalo Nieto 21/02/2024

# Código

**funcions.c**

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

/\* Implementació de les funcions definides al fitxer de capcaleres \*/

int generar\_valor(int min, int max)

{

return min + rand() % (max - min + 1);

}

/\* Fixeu-vos com calculem la posició en la taula 1D \*/

void mostrar\_dades(int \*d, int n\_files, int n\_cols)

{

for (int i = 0; i < n\_files; i++)

{

for (int j = 0; j < n\_cols; j++)

{

printf("%d\t", d[i \* n\_cols + j]);

}

printf("\n");

}

}

bool desar\_taula\_fitxer\_t(int \*d, int n\_files, int n\_cols, char nom[])

{

bool r = false;

FILE \*f;

f = fopen(nom, "w");

if (f != NULL)

{

r = true;

for (int i = 0; i < n\_files; i++)

{

for (int j = 0; j < n\_cols; j++)

{

fprintf(f, "%d\t", d[i \* n\_cols + j]);

// Noteu que necessitem un separador entre valors

}

}

fclose(f);

}

return r;

}

bool desar\_taula\_fitxer\_b(int \*d, int n\_files, int n\_cols, char nom[])

{

bool r = false;

FILE \*f;

f = fopen(nom, "wb"); // Abrimos el archivo en modo binario para escribir

if (f != NULL)

{

r = true;

/\* COMPLETA: Doble bucle for per escriure els valors al fitxer binari

for (int i = 0; i < nf; i++)

{

for (int j = 0; j < nc; j++)

{

fwrite(d, sizeof(int), n\_files \* n\_cols, f);

}

}\*/

// Escribimos los datos en el archivo binario

fwrite(d, sizeof(int), n\_files \* n\_cols, f);

fclose(f);

}

return r;

}

**main.c**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <stdbool.h>**

**#include <time.h>**

**#include "funcions.h" // Funcions pròpies**

**#define MIN\_FC 1**

**#define MAX\_FC 10**

**// Passarem parametres al programa principal**

**int main(int argc, char \*argv[])**

**{**

**int nf, nc; // Nre. de files i columnes que passem com a parametres**

**int \*dades; // La taula de dades, encara sense mida**

**bool hi\_ha\_errors; // Per saber si cal executar o no el programa**

**// Mirar si els parametres son correctes**

**hi\_ha\_errors = false;**

**if (argc == 3)**

**{**

**nf = atoi(argv[1]);**

**nc = atoi(argv[2]);**

**if (nf < MIN\_FC || nf > MAX\_FC || nc < MIN\_FC || nc > MAX\_FC)//Los valores que ha introducido no son correctos**

**{**

**printf("Els valors han d'estar entre %d i %d\n", MIN\_FC, MAX\_FC);**

**hi\_ha\_errors = true;**

**}**

**}**

**else**

**{**

**printf("El nombre de parametres no és correcte!\n");**

**hi\_ha\_errors = true;**

**}**

**// Crear la taula**

**if (!hi\_ha\_errors)**

**{**

**// Reservem espai per la quantitat de bytes que ocupen els elements**

**/\* COMPLETA: Codi per reservar amb malloc() la quantitat de bytes correcta \*/**

**dades = (int \*)malloc(sizeof(int) \* nf \* nc);**

**if (dades == NULL)**

**{**

**printf("Sense espai per la taula!\n");**

**hi\_ha\_errors = true;**

**}**

**}**

**// Emplenar la taula i mostrar-la**

**if (!hi\_ha\_errors)**

**{**

**srand(time(0)); // Inicialitzem la llavor d'aleatoris**

**/\* COMPLETA: emplenar la taula, amb un doble bucle for \*/**

**for (int i = 0; i < nf; i++)**

**{**

**for (int j = 0; j < nc; j++)**

**{**

**dades[i \* nc + j] = generar\_valor(-99999, 99999);**

**}**

**}**

**mostrar\_dades(dades, nf, nc);**

**}**

**// Desar la taula en un fitxer de text**

**if (!hi\_ha\_errors)**

**{**

**if (desar\_taula\_fitxer\_t(dades, nf, nc, "datos\_texto.txt"))**

**{**

**printf("Dades desades correctament en text.\n");**

**}**

**else**

**{**

**printf("Error en desar les dades.\n");**

**}**

**}**

**if (!hi\_ha\_errors)**

**{**

**// Desar la taula en un fitxer binari**

**if (desar\_taula\_fitxer\_b(dades, nf, nc, "datos\_binario.bin"))**

**{**

**printf("Dades desades correctament en binari.\n");**

**}**

**else**

**{**

**printf("Error en desar les dades\n");**

**}**

**}**

**// Alliberem la memòria reservada**

**if (!hi\_ha\_errors)**

**free(dades);**

**printf("Adeu-siau!\n");**

**return 0;**

**}**

**funcions.h**

**/\* Fitxers de capçalera per a inclusio \*/**

**/\*\***

**\* @brief Genera un aleatori entre dos valors**

**\* @param min (E) Valor mínim**

**\* @param max (E) Valor màxim**

**\* @return int**

**\*/**

**int generar\_valor(int min, int max);**

**/\*\***

**\* @brief Mostra en pantalla els valors d'una taula d'enters**

**\* @param d (E) Taula d'enters (punter a enters)**

**\* @param n\_files (E) Nombre de files de la taula**

**\* @param n\_cols (E) Nombre de columnes de la taula**

**\*/**

**void mostrar\_dades(int \*d, int n\_files, int n\_cols);**

**/\*\***

**\* @brief Desa en un fitxer de text els valors d'una taula d'enters**

**\* @param d (E) Taula d'enters (punter a enters)**

**\* @param n\_files (E) Nombre de files de la taula**

**\* @param n\_cols (E) Nombre de columnes de la taula**

**\* @param nom (E) El nom del fitxer**

**\* @return bool cert -> s'ha pogut crear el fitxer, fals -> altrament**

**\*/**

**bool desar\_taula\_fitxer\_t(int \*d, int n\_files, int n\_cols, char nom[]);**

**/\*\***

**\* @brief Desa en un fitxer binari els valors d'una taula d'enters**

**\* @param d (E) Taula d'enters (punter a enters)**

**\* @param n\_files (E) Nombre de files de la taula**

**\* @param n\_cols (E) Nombre de columnes de la taula**

**\* @param nom (E) El nom del fitxer**

**\* @return bool cert -> s'ha pogut crear el fitxer, fals -> altrament**

**\*/**

**bool desar\_taula\_fitxer\_b(int \*d, int nf, int nc, char nom[]);**

# Juego de Pruebas

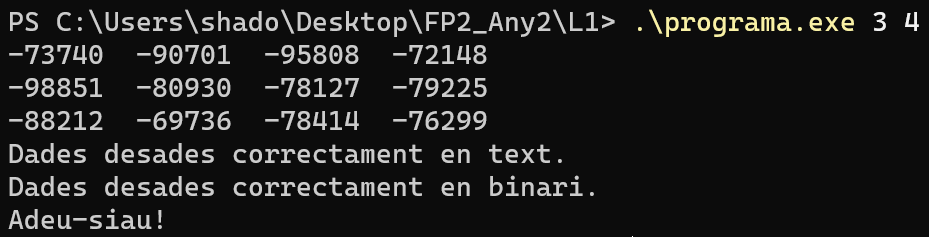
Comando para compilar el programa y crear el ejecutable:

*gcc -o programa main.c funcions.c*

Comando para ejecutar el programa:

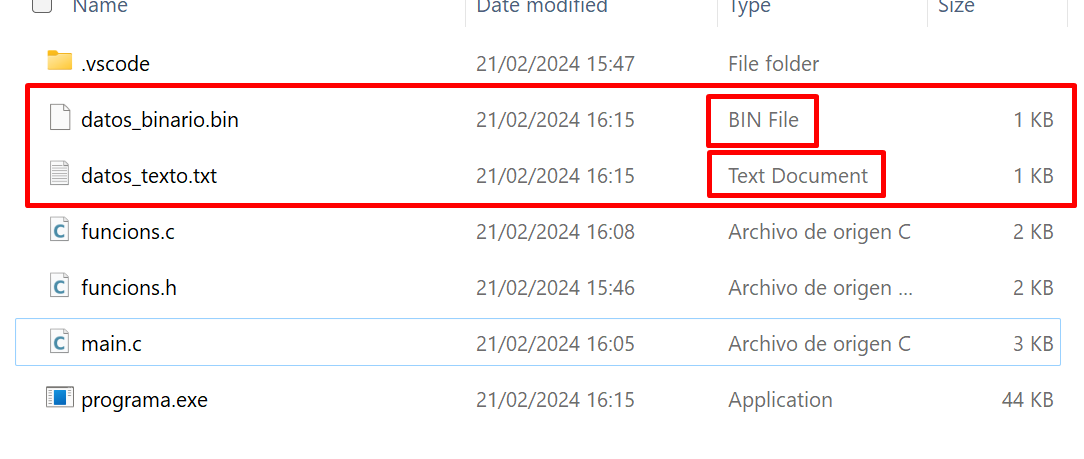
*.\programa.exe valor1 valor2*

Ejemplo 1:

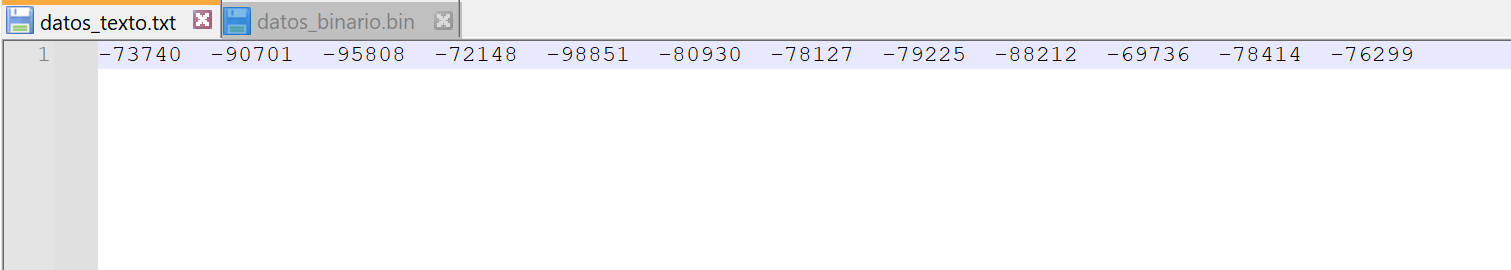


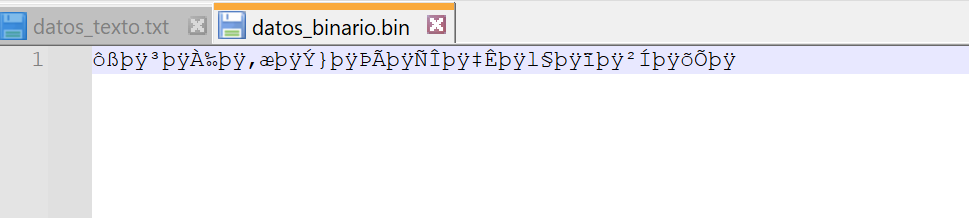
Archivos que se han creado:

Uno de tipo binario y otro de texto.

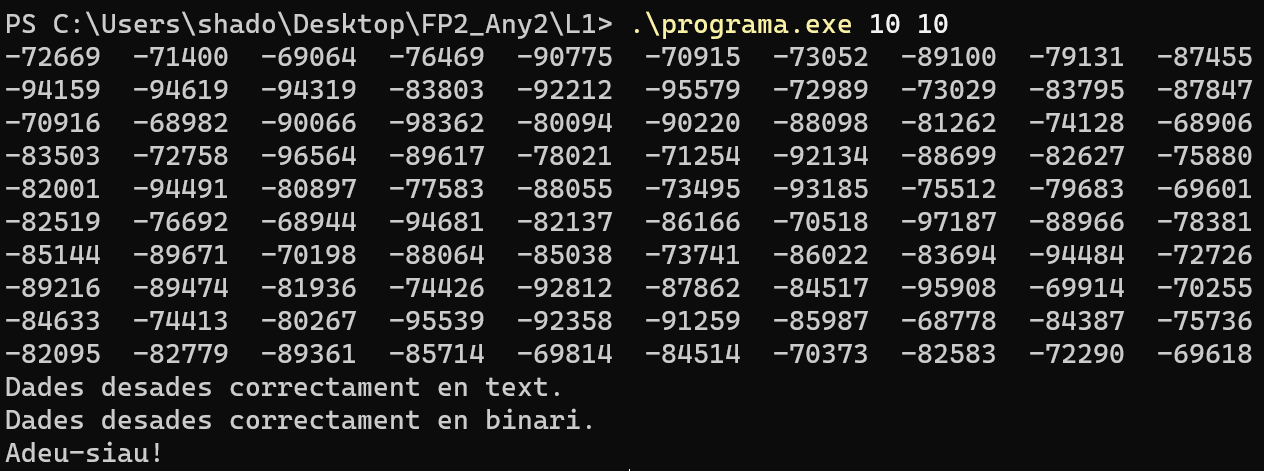


Los abrimos para ver el contenido.El primero es el de tipo texto y el segundo es el de tipo binario:

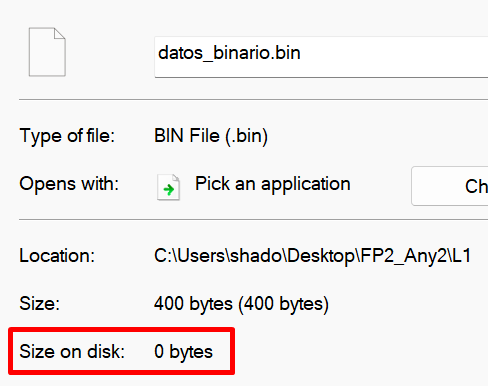




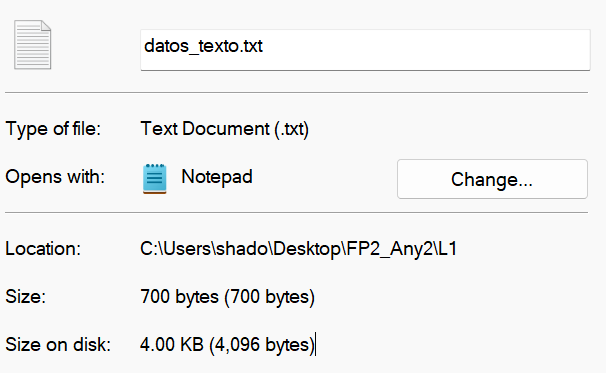
Ejemplo 2:

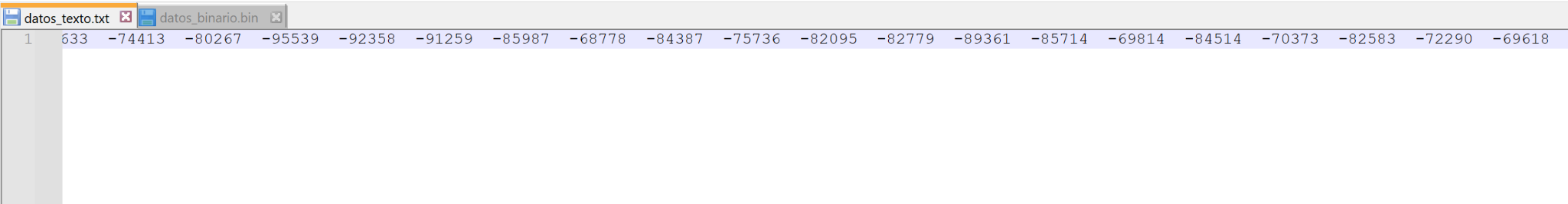


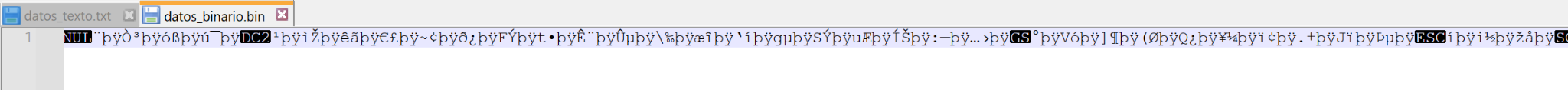
Vemos las propiedades del archivo para tener más detalle sobre este, nos fijamos que el archivo en binario nos indica que no ocupa espacio en el disco:



Pero si vemos el archivo de texto si que ocupa espacio:







Esto se debe a que los archivos binarios pueden contener datos no imprimibles y, por lo tanto, algunos editores de texto o visualizadores de archivos pueden no mostrar correctamente su tamaño total.

Sin embargo, si examinas el tamaño real del archivo binario utilizando herramientas adecuadas, verás que ocupa espacio en disco según la cantidad de datos que contiene.

Es decir los archivos binarios ocupan espacio en el disco, pero su tamaño en pantalla puede no reflejar completamente el tamaño real debido a la forma en que se almacenan los datos binarios y cómo se interpretan al visualizarlos.