



Este trabajo práctico vale 5% de la nota del curso.

Debe ser elaborado individualmente.

No se permite ningún tipo de consulta sobre el contenido de su proyecto a otros estudiantes.

Se debe entregar por BNe a más tardar el 15 de septiembre de 2024

A. OBJETIVOS

- Practicar la representación binaria de diversos tipos de datos.
- Practicar las declaraciones en el lenguaje C y compilar un programa de complejidad pequeña.
- Aprender a usar los editores hexa para manipulaciones de bajo nivel.

B. DESCRIPCIÓN DE LOS EDITORES HEXA

Existen editores (o visores) especializados para diferentes tipos de archivos: texto plano o enriquecido, imágenes, sonido, etc. Pero también existe un tipo especial de editor, el editor hexa, que permite manipular el contenido binario de cualquier tipo de archivo.

Este tipo de editor no le permite visualizar el contenido —sea este una imagen, un texto o cualquier otra cosa— sino la representación binaria de la entidad en el archivo. Por supuesto, para manipular el contenido es necesario conocer el formato —la estructura de datos— del archivo en cuestión.

Para mayores detalles pueden consultar:

https://en.wikipedia.org/wiki/Hex_editor

http://www.linfo.org/hex_editor.html

Y en el siguiente enlace una explicación más detallada de su manejo:

https://www.sweetscape.com/articles/hex_editor.html

Hay diferentes editores hexa, en realidad no importa cuál use (¡todos manejan binario físico, así que da lo mismo!), pero hay uno llamado HxD que es simple ligero y libre.

Puede descargarlo de: <https://mh-nexus.de/en/hxd/>

Instálelo en su máquina virtual, y realice las siguientes actividades.

ATENCIÓN: si usa el *data inspector* asegúrese de seleccionar la opción “*big endian*”.

C. DESCRIPCIÓN DEL PROBLEMA

Usando el editor hexa, cree un archivo binario con el siguiente contenido:

- 1- En los primeros cuatro bytes, escriba su código de estudiante como un número binario (por supuesto, en el editor lo debe teclear en hexa).
- 2- En los siguientes 70 bytes, escriba sus nombres, en formato de *string*, separados por espacios. Después de todos los nombres, debe haber una coma seguida de una tabulación (TAB), y luego los apellidos separados por espacio. Al final, un

cambio de línea (carácter de control CR seguido de LF); por último, uno o más caracteres NUL hasta completar los 70 bytes. Nota: Los tres últimos caracteres (CR, LF y NUL) siempre deben estar, y el total de bytes no puede diferir de 70 (si es necesario trunque nombres y/o apellidos).

- 3- Luego, en los siguientes 13 bytes, escriba su cédula como una *string* de dígitos decimales, es decir, usando los caracteres-dígitos decimales del código ASCII. Al final también debe haber un cambio de línea (CR seguido de LF). y un carácter NUL.
- 4- Después, escriba la fecha de su cumpleaños en el siguiente formato: un byte para el día, y luego 2 bytes para el mes y el año (los dos juntos: los 4 bits más significativos para el mes y los 12 restantes para el año); todos son números positivos.
- 5- Después, también como un número binario pero esta vez de dos bytes, escriba, como un número negativo en complemento a dos, cuánto hay que restarle al año de su nacimiento para obtener el año de su nacimiento pero en el calendario islámico (Hijiri; en el web encuentra conversores de fecha gregoriana a islámica)

ATENCIÓN: debe escribir su nombre y código REALES. Los demás datos los puede poner inventados si así lo desea (¡pero deben ser realistas!: no tiene 80 ni mil años, y las cédulas son de 10 dígitos). Los puntos 4 y 5 deben ser coherentes.

Compile el programa validador y úselo sobre el archivo:

Para eso, use el programa validador en C adjunto.

“Validar”, en este contexto, quiere decir que revisa el formato de lo que usted realizó y lo imprime en pantallas como ASCII o valores decimales para que usted pueda revisarlos.

Tenga en cuenta que el programa no puede verificar todo (¡el programa no sabe cuál es su cédula!), solo se limita a imprimir los valores del archivo según el formato y tipo especificados. Usted debe verificar que el valor sea efectivamente el que quería poner, así como la coherencia entre los campos.

ATENCIÓN: el programa no está completo. Al comienzo tiene una estructura (struct Datos) con un campo para cada dato del archivo (y en el mismo orden). Pero estos campos están declarados sin tipo de datos. Usted debe:

1. completar las declaraciones.
2. compilar y ejecutar su programa sobre su archivo de datos.

```
C:\> Símbolo del sistema
D:\Google Drive\InfraTec\Evaluaciones\2023-1-1304 TP1>main p3.bin
Codigo:151110039
Nombres:Theophrastus Phillippus Aureolus, Bombastus von Hohenheim
Cedula:1234567890
Dia:10
Mes-Año:11-1493
Diferencia anyo islamico:-594
```

Ejemplo del uso y salida del programa
En este ejemplo el programa se llama “main” y el archivo, “p3.bin”.

Tenga presente que las declaraciones se deben completar con el tipo de datos más apropiado para el valor en cuestión; esto incluye el tamaño y el signo.

D. CONDICIONES DE ENTREGA

- Entregar el archivo de datos binarios, el código fuente del programa en C con las declaraciones completas y su respectivo ejecutable en un archivo *.zip. **El nombre del archivo debe ser: TP1_código_apellido_nombre.zip. Al comienzo del archivo fuente escriba su nombre, código y correo (ver esqueleto). No cumplir con las anteriores condiciones tendrá una penalidad del 10% de la nota.** Si su programa no funciona o si tiene particularidades, puede enviar un archivo .docx o .pdf explicando por qué cree que no funciona o qué fue lo que hizo.
- Los programas se calificarán únicamente usando el ambiente de visual de las máquinas virtuales. Si el programa no compila en este ambiente, se considerará que no corre (así compile en otros ambientes).
- El trabajo se realiza individualmente. No debe haber ninguna consulta sobre el contenido con otras personas.
- De ser necesario se puede solicitar una sustentación. Dicha sustentación puede afectar la nota.
- **Entregar por BNe a más tardar el 15 de septiembre de 2024 hasta las 11:50 pm.**

E. CRITERIOS DE CALIFICACIÓN PARA LOS PROGRAMAS

La calificación consta de dos partes:

- Correcta representación de los datos en binario (80%).
- Correcta declaración de los campos en C (20%).