



Este proyecto vale 5% de la nota del curso.

Debe ser elaborado individualmente.

No se permite ningún tipo de consulta sobre el proyecto con otras personas.

Se debe entregar por Sicua+ a más tardar el **5 de noviembre de 2021**

#### A. OBJETIVOS

- Practicar el lenguaje C y desarrollar un programa de baja complejidad.
- Conocer las operaciones de C para el manejo de bits.
- Aplicar lo anterior descomponiendo un vector de enteros en bytes.

#### B. DESCRIPCIÓN DEL TRABAJO

El objetivo del proyecto es desarrollar un programa en C que lea un vector de enteros e imprima cada uno de sus elementos descomponiéndolos en los bytes que lo conforman. El vector puede tener cualquier tamaño.

Se recorre el vector desde el comienzo y cada uno de sus elementos se imprime en una línea, byte por byte, en hexa, de más significativo a menos significativo y separados por tabulaciones.

En cada línea se debe identificar el elemento que se está imprimiendo ( $v[i]$ )

##### Ejemplo:

El usuario indica que  $n = 2$ , y teclea los elementos: 1215261793 y 46

##### Vector de int (tamaño 2):

	+ <i>significativo</i> -				+ <i>significativo</i> -			
Valor en hexa	0x48	0x6F	0x6C	0x61	0x00	0x00	0x00	0x2E
Valor en decimal	1.215'261.793				46			

##### Imprime:

$v[0] =$         0x48   0x6F   0x6C   0x61

$v[1] =$         0x00   0x00   0x00   0x2E

##### Estructuras de datos

- Una variable para almacenar el tamaño del vector.
- Un apuntador a int: para apuntar al vector de enteros.

## Estructura del programa

El programa solo debe tener dos procedimientos: el main y otro procedimiento encargado de imprimir el vector.

### *main:*

- Le pide al usuario el tamaño del vector para inicializar la respectiva variable (validar que  $n > 0$ ).
- Declara e inicializa el apuntador al vector de int. La inicialización consiste en ponerlo a apuntar a un vector de int del tamaño pedido. El vector se crea dinámicamente usando la función de C calloc.
- Inicializa el vector: le pide al usuario que vaya tecleando cada uno de los elementos solicitados.
- Invoca el procedimiento que imprime el vector.
- Imprime el resultado (el vector de int) byte por byte en hexa.

### *Procedimiento de empaquetamiento:*

- Tiene dos parámetros: el apuntador al vector de enteros y un entero con el tamaño.
- Recorre el vector, elemento por elemento, imprimiéndolos en hexa. Uno por línea descompuesto en sus bytes de más a menos significativo.

## Restricciones y consideraciones

- Las estructuras de datos se declaran e inicializan en el main.
- El programa solo debe constar del main y el procedimiento de cálculo.
- No puede usar librerías externas que resuelvan directamente el problema; solo las básicas de C (entrada/salida, manejo de cadenas, etc.).
- NO PUEDE usar el formato %x de printf.
- Los dígitos hexa deben ir en letras mayúsculas y precedidos por los caracteres "0x".
- El vector puede tener cualquier tamaño.
- 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).

## C. CONDICIONES DE ENTREGA

**Entregar el código fuente junto con el ejecutable en un archivo .zip. TP3\_NOMBRES\_APELLIDOS.zip y al comienzo del archivo fuente escriba su nombre, código y correo.**

Si su programa no funciona o si su solución tiene particularidades, puede enviar un archivo .docx o .pdf explicando por qué cree que no funciona o qué fue lo que hizo.

El trabajo es individual. No debe haber consultas con otros estudiantes.

Se puede solicitar una sustentación sobre cualquier parte del trabajo. Dicha sustentación puede afectar la nota.

El proyecto debe ser entregado por **Bloque Neón** a más tardar el **5 de noviembre de 2021 hasta las 11:50 pm.**

#### D. CRITERIOS DE CALIFICACIÓN PARA LOS PROGRAMAS

La calificación consta de dos partes:

- Ejecución (50%). Se harán 5 pruebas (los dos casos de prueba y otras 3 adicionales). Para cada caso, se revisará si la salida es correcta o no según los requerimientos establecidos en el enunciado. Cada prueba vale 10%.
- Inspección del código (50%). Se consideran tres aspectos:
  - o 10% - legibilidad (nombres dicientes de variables, comentarios, indentación)
  - o 20% - manejo de bits (uso correcto y eficaz de los operadores de bits de C: >>, &, etc. para resolver el problema)
  - o 20% - Correcto y eficaz manejo de la estructura de datos (recorrido, manejo de los elementos).
- Penalidad -10%, en caso de no cumplir con la condición de entrega: **Entregar el código fuente junto con el ejecutable en un archivo .zip. TP3\_NOMBRES\_APELLIDOS.zip y al comienzo del archivo fuente escriba su nombre, código y correo.**

#### E. CASOS DE PRUEBA

##### Caso 1:

Entrada: n = 1; 4128804

Salida:

v[0] =        0x00   0x3F   0x00   0x24

##### Caso 2:

Entrada: n = 3; -2147483648, 0, 2,147,483,647

v[0] =        0x80   0x00   0x00   0x00

v[1] =        0x00   0x00   0x00   0x00

v[2] =        0x7F   0xFF   0xFF   0xFF