



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 BNe a más tardar el **26 de octubre de 2022**

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 haciendo una rotación de bits en un vector de enteros.

B. DESCRIPCIÓN DEL TRABAJO

El objetivo del proyecto es desarrollar un programa en C que lea un vector de enteros de tamaño n para después realizar una rotación a la izquierda de tamaño k usando el vector completo como operando.

Nota: en este artículo puede ver la definición de la operación de rotación: https://es.wikipedia.org/wiki/Operador_a_nivel_de_bits

El programa toma todo el vector como un solo número y hace una rotación a la izquierda de k posiciones.

El vector de enteros puede tener cualquier tamaño; de hecho, el programa debe empezar preguntándole dicho tamaño (n) al usuario, para después proceder a leer los n elementos que lo componen. También debe pedir el valor de k .

Al final imprime el vector, un elemento por línea, en hexadecimal (ver formato %X de printf). Cada elemento debe venir precedido por " $v[i]$ " donde " i " es su subíndice respectivo.

Ejemplo:

El usuario indica que $n = 2$, y después ingresa los valores: -2 y 12. Adicionalmente indica que $k = 4$.

El vector de int (tamaño 2) es:

	+	significativo	-	+	significativo	-
Valor en hexa	0xFF	FF	FF	FE	0x00	00 00 0C

Después de la rotación, el vector debe ser:

	+	significativo	-	+	significativo	-
Valor en hexa	0xFF	FF	FF	E0	0x00	00 00 CF

Imprime:

$v[0] = 0xFFFFFEE0$

`v[1] = 0x000000CF`

Estructuras de datos

- Unas variables para almacenar el tamaño del vector (n) y el número de corrimientos (k).
- 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 efectuar la rotación.

main:

- Le pide al usuario el tamaño del vector (n) 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 teclee el valor de cada elemento.
- Le pide al usuario el número de posiciones (k) de la rotación (validar $0 < k \leq 32$).
- Invoca el procedimiento que realiza la rotación en el vector.
- Imprime el resultado (el vector de `int`) posición por posición en hexadecimal.

Procedimiento de rotación:

- Tiene tres parámetros: un apuntador al vector de enteros, un entero con el tamaño del vector y un entero con el valor de k.
- Realiza la rotación sobre el vector.

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.).
- Los números 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`. El nombre del archivo debe ser: `TP3_código_apellido_nombre.zip`

Al comienzo del archivo fuente escriba su nombre, código y correo.

No cumplir con las anteriores condiciones tendrá una penalidad del 10% de la nota.

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 BNe a más tardar el **26 de octubre de 2022 hasta las 11:50 pm. No se recibirán trabajos entregados con posterioridad.**

D. CRITERIOS DE CALIFICACIÓN PARA LOS PROGRAMAS

La calificación consta de dos partes:

- Ejecución (50%). Se harán 5 pruebas (los tres casos de prueba y otras 2 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 (declaración, creación, recorrido, manejo de los elementos, lectura, impresión).

No cumplir con las condiciones de entrega acarreará una penalidad del 10% de la nota.

E. CASOS DE PRUEBA

Caso 1:

Entrada: $n = 2$; elementos = -2, 12; $k = 8$

Salida:

$v[0] = 0xFFFFFFFFE00$

$v[1] = 0x00000CFF$

Caso 2:

Entrada: $n = 3$; elementos = -2, 12, 0; $k = 32$

$v[0] = 0x0000000C$

$v[1] = 0x00000000$

$v[2] = 0xFFFFFFFFFE$

Caso 3:

Entrada: $n = 2$; elementos = -2, 12; $k = 3$

Salida:

$v[0] = 0xFFFFFFFFF0$

$v[1] = 0x00000067$