

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI3641 – Lenguajes de Programación I
Septiembre–Diciembre 2025 Examen 2

Examen 3 (20 puntos)

Estudiante Rafael Antonio Valera Pacheco 16-11202

Video de los capítulos: <https://youtu.be/cQ20I-ObKDc>

Capítulo 7 — Sección 7.1

Capítulo 7 — Sección 7.2

Capítulo 7 — Sección 7.4

Capítulo 8 — Sección 8.1 a 8.2

Capítulo 8 — Secciones 8.4 a 8.6

Parte escrita

1. Tomando en cuenta las definiciones de X, Y y Z planteadas en los párrafos de introducción del examen, considere las siguientes definiciones de variables:

$$L_1 = \min(X, Y) \quad U_1 = \max(X, Y) + 1$$

$$L_2 = \min(X, Z) \quad U_2 = \max(X, Z) + 1$$

$$L_3 = \min(Y, Z) \quad U_3 = \max(Y, Z) + 1$$

$$I = \left\lfloor \frac{L_1 + U_1}{2} \right\rfloor \quad J = \left\lfloor \frac{L_2 + U_2}{2} \right\rfloor \quad K = \left\lfloor \frac{L_3 + U_3}{2} \right\rfloor$$

Considere también la siguiente declaración:

$M : \text{array } [L_1..U_1] \text{ of array } [L_2..U_2] \text{ of array } [L_3..U_3] \text{ of } T$

Suponiendo que M inicia en la dirección cero (0) y que el tamaño del tipo T es cuatro (4), se desea que calcule:

- La dirección de $M[I][J][K]$ si las matrices se guardan en *row-major*.
- La dirección de $M[I][J][K]$ si las matrices se guardan en *column-major*.

Para resolver este planteamiento comenzaré hallando los datos necesarios para poder responder las partes a y b, entonces comenzamos con qué valores tiene X, Y y Z los mismo son X=2, Y=0 y Z=2.

Paso 1: Hallar el valor de los Lsub(?) y Usub(?)

Entonces:

$$L_1 = \min(2, 0) = 0 \text{ y } U_1 = \max(2, 0) + 1 = 3$$

Rango: [0..3] —> Tamaño (D1) = 4

$$L_2 = \min(2, 2) = 2 \text{ y } U_2 = \max(2, 2) + 1 = 3$$

Rango: [2..3] —> Tamaño (D2) = 2

$$L_3 = \min(0, 2) = 0 \text{ y } U_3 = \max(0, 2) + 1 = 3$$

Rango: [0..3] —> Tamaño (D3) = 4

Paso 2: Ya que tengo los datos que necesito paso a encontrar los valores de I,J y K. Además el desplazamiento que necesitaré más adelante.

$$I = [(0 + 3)/2] = [1.5] = 1$$

Desplazamiento Isubeff: $I - L_1 = 1 - 0 = 1$

$$J = [(2 + 3)/2] = [2.5] = 2$$

Desplazamiento Jsubeff: $J - L_2 = 2 - 2 = 0$

$$K = [(0 + 3)/2] = [1.5] = 1$$

Desplazamiento Ksubeff: $K - L_3 = 1 - 0 = 1$

Paso 3: Calcular el direccionamiento de memoria y también los tamaños, afortunadamente ya el enunciado nos dice que supongamos ciertos valores.

Dirección Base (BA) = 0

Tamaño (S) = 4

Paso 4: Ya con todos los datos encontrados puedo responder el apartado a y b.

(a) Row Major

Para encontrar la respuesta hay que saltar bloques enteros de la Dimensión 1 luego saltar filas de la Dimensión 2 y la posición en Dimensión 3. Todo esto nos da una fórmula general que es:

$$Dir = BA + S \times [I_{subeff} \times D2 \times D3) + (J_{subeff} \times D3) + K_{subeff}]$$

Sustituimos valores:

$$Dir = 0 + 4 \times [8 + 0 + 1] = 36$$

Finalmente la respuesta a el apartado a es 36.

(b) Column Major

Para este apartado es la misma lógica solo que a la inversa por lo que la fórmula cambia un poco en el orden de sus términos

$$Dir = BA + S \times [(K_{subeff} \times D2 \times D1) + (J_{subeff} \times D1) + I_{subeff}]$$

Sustituimos valores:

$$Dir = 0 + 4 \times [8 + 0 + 1] = 36$$

Finalmente la respuesta a el apartado b es 36.

Para concluir la razón por la cual los dos resultados me dieron lo mismo, fue por una coincidencia producto de los números de mi carnet en el procedimiento y esto es porque en Row-major saltamos $1 \times (2 \times 4) = 8$ bloques. En Col-major saltamos $1 \times (4 \times 2) = 8$ bloques. Como el índice del medio J tuvo un desplazamiento de 0, no generó diferencia entre los métodos.

Se desea que modele e implemente, en el lenguaje de su elección, un programa que simule un manejador de tipos de datos. Este programa debe cumplir con las siguientes características:

- (a) Debe saber manejar tipos atómicos, registros (struct) y registros variantes (unión).
- (b) Una vez iniciado el programa, pedirá repetidamente al usuario una acción para proceder. Tal acción puede ser:

Tal acción puede ser:

I. ATOMICO <nombre><representación><alineación>

Define un nuevo tipo atómico de nombre <nombre>, cuya representación ocupa <representación> bytes y debe estar alineado a <alineación> bytes.

Ejemplo: ATOMICO char 1 2 y ATOMICO int 4 4

II. STRUCT <nombre> [<tipo>]

Define un nuevo registro de nombre <nombre>. La definición de los campos del registro viene dada por la lista en [<tipo>]. Nótese que los campos no tendrán nombres, sino que serán representados únicamente por el tipo que tienen.

Ejemplo: STRUCT foo char int

III. UNION <nombre> [<tipo>]

Define un nuevo registro variante de nombre <nombre>. La definición de los campos del registro variante viene dada por la lista en [<tipo>].

Ejemplo: UNION bar int foo int

IV. DESCRIBIR <nombre>

Debe dar la información correspondiente al tipo con nombre <nombre>. Esta información debe incluir tamaño, alineación y cantidad de bytes desperdiciados para el tipo, bajo distintas estrategias:

- Sin empaquetar.
- Empaquetado.
- Reordenando los campos de manera óptima.

V. SALIR

Debe salir del simulador.

El programa deberá pedir la siguiente acción al usuario después de cada ejecución. Además, debe incluir pruebas unitarias con cobertura mayor al 80 %.

Código en Github:

<https://github.com/Moepiii/Teoria-de-lenguajes-de-programacion/tree/main/Tarea%20o%20Examen%203%20Rafael%20Valera%201611202>