

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Cursó con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, C++, o en Pascal, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros; en Pascal el concepto prototipo es análogo al encabezado del procedimiento o de la función previa al bloque que lo define.

### Cadena de Hipermercados

*Temas evaluados: Resolución de problemas, estructuras de datos, y lenguaje de programación*

#### Contexto

Usted es miembro de un equipo de desarrollo de software. El equipo debe desarrollar un sistema para **gestionar las ofertas en las diferentes sucursales de una cadena de hipermercados**. Parte del equipo se encarga del almacenamiento de las ofertas en disco, a usted le toca la **interacción con el usuario y la actualización de los ofertas en memoria**.

#### Problema

Desarrollar un programa que permita **listar las ofertas, agregar una oferta, quitar una oferta, o quitar todas las ofertas**. Las actualizaciones pueden aplicarse a **una sucursal** en especial o a **todas**. Las sucursales se identifican con  $N \in \mathbb{N}, 1 \leq N \leq 5$ ; las ofertas se codifican con enteros de 7 dígitos decimales.

*Se pide para los temas 1 y 2*

1. **Codifique** las declaraciones para implementan la estructura de datos que almacene los códigos de todas las ofertas de cada una de las  $N$  sucursales.
2. **Diseñe o codifique** la función **Interactuar**. Implemente el menú de acciones con un ciclo **while**, el ciclo finaliza cuando el usuario selecciona la opción **Finalizar**.  
Para obtener la selección del usuario, invoque a **GetAccion**, que retorna **Listar, Insertar, Eliminar, Vaciar, o Finalizar**. Invoque a **GetEspacioDeAccion** para determinar el alcance de las acciones; la función retorna cero para indicar que la acción es para todas las sucursales, si no, retorna un valor entre 1 y  $N$  para indicar que aplica a una sucursal determinada.  
Note que si la acción elegida es Insertar o Eliminar, es necesario solicitar al usuario la oferta.  
Para realizar cada acción, la función Interactuar debe invocar, con los parámetros adecuados, a las funciones **Listar, Insertar, Eliminar, y Vaciar**.

3. **Codifique** los prototipos de:

- 3a. **GetAccion**.
- 3b. **GetEspacioDeAccion**.
- 3c. **Listar**.
- 3d. **Insertar**.
- 3e. **Eliminar**.
- 3f. **Vaciar**.

Considere que las últimas cuatro funciones y pueden aplicar a una o a todas las sucursales, en función de **GetEspacioDeAccion**.

*Tema 1, para quienes tienen legajo con dígito verificador (el último dígito) menor a 5.*

4. **Diseñe o codifique** la función **Eliminar** una oferta.

*Tema 2, para quienes tienen legajo con dígito verificador (el último dígito) mayor a 4.*

4. **Diseñe o codifique** la función **Insertar** una oferta.

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Cursó con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, C++, o en Pascal, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros; en Pascal el concepto prototipo es análogo al encabezado del procedimiento o de la función previa al bloque que lo define.

### Validar Expresiones

*Temas evaluados: Resolución de problemas, tratamiento de secuencias, y lenguaje de programación*

#### Problema

Validar que una expresión aritmética esté **bien formada**. Las expresiones son simples, y están compuestas por **cuatro operadores**: +, -, \*, y /; los operandos son **letras minúsculas** que representan variables; puede haber **paréntesis**, **corchetes**, o **llaves** para cambiar la precedencia.

Ejemplo de expresiones correctas:

```
a+b*c
((a+b)*(c+d))
[(a+b)*(c+d)]
```

Ejemplo de expresiones incorrectas:

```
a+*bc
(a+b)*c+d+
[(a+b)*(c)]+d)
```

1a. **Escriba** el prototipo de **Validar** que retorna verdadero si la cadena dato es una expresión válida.

1b. **Diseñe** o codifique la función **Validar**. Invoque las funciones de pilas para verificar el balanceo de los **paréntesis**

Ayuda:

Al encontrar paréntesis, corchete o llave izquierda, colocarlo en una pila

Al encontrar paréntesis, corchete o llave derecha saque el tope de la pila, debe ser el izquierdo que se corresponda para verificar el balanceo.

Al final del proceso, para controlar que efectivamente hayan balanceado, como cree que debe quedar la pila?

Se supone que la cadena solo tiene caracteres validos, estos son +, -, \*, /, (, ), {, }, [, ] y letras minúsculas

Para controlar verifique que:

- Si anterior ParIzq, entonces actual no debe ser operador ni ParDer;
- Si anterior Letra, entonces actual no debe ser Letra ni ParIzq;
- Si anterior Operador, entonces actual no debe ser operador ni ParDer;
- Si anterior ParDer, entonces actual no debe ser letra, ni ParDer
- La expresión no debe terminar con un operador

Puede utilizar, si lo desea, alguna de las siguientes funciones sin desarrollar:

**char caracterOpuesto(char c)** si se invoca con un paréntesis, corchete o llave derecho, retorna el equivalente izquierdo

**bool esOperador(char c)** retorna true si se invoca con uno de los operadores validos

**bool esDerecho(char c)** retorna true si se invoca con ), ] o }

**bool esIzquierdo(char c)** retorna true si se invoca con (, [ o {

**bool esLetra(char c)** retorna true si se invoca con una letra a..z

Apellido y nombre: \_\_\_\_\_ Legajo: \_\_\_\_\_ Curso con Prof: \_\_\_\_\_

Cantidad de hojas entregadas: \_\_\_\_\_ Nota: \_\_\_\_\_ Evaluó Prof: \_\_\_\_\_

- Si luego de la lectura del examen, durante la resolución tiene alguna duda, escriba hipótesis de trabajo, las cuales también serán evaluadas.
- Los puntos que solicitan codificación puede ser respondidos en C, C++, o en Pascal, pero debe indicar el lenguaje utilizado.
- En C y C++ prototipo refiere a la declaración de la función, es decir tipo de dato retornado, nombre de la función, y tipos de los parámetros; en Pascal el concepto prototipo es análogo al encabezado del procedimiento o de la función previa al bloque que lo define.

## Implementación de Matrices

*Temas evaluados: Resolución de problemas, abstracción de datos, estructuras enlazadas, y lenguaje de programación*

### Problema

Se requiere desarrollar cinco operaciones sobre matrices de enteros de N filas  $\times$  M columnas representadas con estructuras enlazadas:

- Crear la matriz a partir de N y M (dados). Crea la matriz representada con estructuras enlazadas.
- Asignar un entero a la fila i y columna, j. Asigna un valor al nodo correspondiente.
- Obtener el entero de la fila i, columna j. Obtiene el valor almacenado en el nodo correspondiente.
- Sumar dos matrices. Asume que las matrices son del mismo tamaño.  $M1_{ij} + M2_{ij} = M3_{ij}$
- Producto escalar. Multiplicar cada elemento de la matriz por un entero dado.  $k \cdot M4_{ij} = (k \cdot M5)_{ij}$

1. Codifique la declaración de una estructura de datos enlazada que implemente las matrices.

2. Escriba el prototipo de las cinco operaciones.

*Tema 1*, para quienes tienen legajo con dígito verificador (el último dígito) menor a 5.

**3a.** Diseñe o codifique el subprograma para la operación ii. Asignar.

**3b.** Diseñe o codifique el subprograma para la operación iv. Sumar.

*Tema 2*, para quienes tienen legajo con dígito verificador (el último dígito) mayor a 4.

**3a.** Diseñe o codifique el subprograma para la operación iii. Obtener.

**3b.** Diseñe o codifique el subprograma para la operación v. Producto escalar.

**4.** Puntos extras: Diseñe o codifique la función *operarMatriz* que dada una matriz M1 (representada con estructura enlazada), y un escalar retorne la matriz M2 producto de la original por el escalar y la matriz M3 suma de la original más la obtenida en la multiplicación:

M1 y k      La matriz original y el escalar

$M2 = k * M1$       M2 la matriz producto de M1 por el escalar. Recuerde  $M2[i][j] = k * M1[i][j]$ .

$M3 = M1 + M2$ . M3 la matriz suma de M1 + M2. Recuerde  $M3[i][j] = M1[i][j] + M2[i][j]$ .

Nota:

1. La matriz M1 debe estar representada por una estructura enlazada. Para la matriz M2 y M3 decida la estructura abstracta que prefiera y justifique brevemente su elección.
2. Criterio de corrección: La correcta resolución de los puntos 1,2 y 3 garantiza aprobación, el agregado del punto 4 mejora la evaluación.