

202475 - EXAMEN
PROGRAMACIÓN
INGENIERÍA CIVIL EN INFORMÁTICA Y
TELECOMUNICACIONES

Nombre Completo: _____

Puntaje Obtenido: _____ / 100

Instrucciones

1. Esta evaluación es individual, por lo tanto, cualquier situación que se pueda interpretar como intento de copia se penalizará con la nota mínima.
2. Solo se responderán dudas del enunciado durante los primeros 10 minutos de la evaluación. Cualquier duda con respecto al desarrollo no será respondida.
3. Está prohibido separar las hojas de esta evaluación, el uso de celular, apuntes, cuadernos, computadores o cualquier otro dispositivo durante el transcurso de la prueba. Sin embargo, puede utilizar los archivos PDF de las clases.
4. Para esta evaluación se debe utilizar lápiz pasta azul o negro, en caso de usar lápiz mina no se revisará su evaluación posterior a la entrega de notas (no tendrá derecho a apelación).
5. Solo se deben utilizar contenidos vistos en clase, es decir, el uso de comandos o funciones no aprobadas por el docente implican nota mínima para la presente evaluación.
6. Deberá entregar su evaluación antes de la hora de término indicada por el docente/ayudante.

Enunciado N°1 (30 puntos)

La suma algebraica de dos o más polinomios consiste en agrupar sus términos semejantes para luego sumarlos según sus signos y formar un único polinomio con el resultado de la operación.

Por ejemplo, para sumar polinomios de variable X es necesario reconocer cuáles son los términos semejantes, es decir, aquellos que tienen igual exponente. Luego de esto, tales términos deben ser agrupados y sus coeficientes sumados o restados, según corresponda.

Supongamos que es necesario sumar dos polinomios de grado 2. Siguiendo la regla anterior, es posible identificar sus términos semejantes para luego encontrar el resultado de la suma de sus coeficientes, tal como se muestra a continuación:

$$(-6x^2 + 9x - 6) + (3x^2 - 4) = -3x^2 + 9x - 10$$

Por otro lado, en caso de requerir el producto de un polinomio por un número, sólo es necesario multiplicar cada coeficiente del polinomio por el número, tal como se muestra a continuación:

$$3 \cdot (2x^3 - 3x^2 + 4x - 2) = 6x^3 - 9x^2 + 12x - 6$$

Ahora que sabemos cómo sumar polinomios de una variable y multiplicarlos por un número, debemos saber que un polinomio puede ser representado como una lista invertida. Para ello, se puede utilizar el índice como el grado de cada término y el contenido asociado a ese índice puede corresponder al coeficiente, tal como ocurriría en el ejemplo de suma de polinomios:

Índices	0	1	2
A	-6	9	-6
B	-4	0	3
Resultado Suma	-10	9	-3

Considerando la información entregada, construir el siguiente programa:

1. Crear una función llamada `crear_polinomio(grado)` que reciba el grado máximo de un polinomio y devuelva una lista con números enteros ingresados por el usuario para los coeficientes de cada término desde 0 hasta el grado indicado, la cual debe ser devuelta por la función. (10 puntos)
2. Crear una función llamada `sumar_polinomios(polimonioA, polimonioB)` que reciba dos polinomios de igual grado y devuelva una lista con el polinomio resultante luego de sumar los términos de los polinomios recibidos. (10 puntos)
3. Crear una función llamada `multiplicar_polinomio(polimonio, numero)` que reciba un polinomio y un número entero, y devuelva una lista con el polinomio resultante luego de multiplicar los términos del polinomio y el número recibidos. (10 puntos)

Enunciado N°2 (35 puntos)

La sala de cine “Don Pepito” tiene $n \times m$ asientos (n filas y m columnas) para sus presentaciones. Esta sala tiene sus asientos dispuestos como una matriz y sus administradores requieren un programa que les permita administrarlos. Por último, cada asiento es identificado por una fila y una columna y tales asientos pueden estar ocupados (1) o desocupados (0).

Matriz ejemplo:

	Pantalla			
0	1	1	0	0
1	1	0	0	1
2	1	0	0	1
3	1	1	0	1
4	1	0	1	0
	0	1	2	3

- (10 puntos) Crear una función llamada `comprar_asiento(matriz_asientos, fila, columna)` que reciba la fila y columna que se desea reservar y valide si el asiento solicitado se encuentra desocupado. De ser así, se debe marcar el asiento como ocupado. En caso de que el asiento esté ocupado, se debe asignar un asiento distinto que se encuentre vacío e imprimir la posición. En caso de que no haya un asiento vacío, se debe que se asignó un nuevo asiento e indicar las coordenadas. En cualquier caso, se debe retornar la matriz de asientos actualizada.
- (10 puntos) Crear una función llamada `calcular_uso(matriz_asientos)` que reciba la matriz de asientos, determine la cantidad de asientos ocupados y desocupados y devuelva el % de asientos ocupados.
- (15 puntos) Crear una función llamada `calcular_ventas(matriz_asientos)` que reciba la matriz de asientos y calcule el total de ventas considerando los siguientes precios por asiento ocupado según la fila en que se encuentra:

	Precio Unitario
Fila 0	100
Filas 1 y 2	50
Otras Filas	20

Enunciado N°3 (35 puntos)

Una empresa relacionada con el rubro automotriz necesita mantener un control de los vehículos de su propiedad que se encuentran en taller para reparación. Para esto, hace uso de un archivo llamado `vehículos.csv` en el cual se almacena la marca y modelo de todos los vehículos de la compañía. Con esta información se debe construir un programa que cumpla con lo siguiente:

1. (10 puntos) Crear una función llamada `obtener_datos(ruta_archivo)` que reciba la ruta del archivo `vehículos.csv` con registros de marcas y modelos de vehículos, y retorne una lista con los registros de la marca y modelo de cada vehículo.
2. (10 puntos) Crear una función llamada `mostrar_numero_vehiculos_por_marca(lista_vehiculos)` que reciba una lista con vehículos y retorne otra lista con el número de vehículos por cada marca. Cada marca sólo puede aparecer 1 vez en la lista retornada.
3. (15 puntos) Crear una función llamada `guardar_vehiculos_por_marca(lista_vehiculos)` que reciba una lista con vehículos y cree un archivo `.csv` por cada marca. Cada uno de los archivos debe tener por nombre la marca (Ej: `Audi.csv`) y contener todos los modelos de esa marca bajo el encabezado `Modelo`. Si, por ejemplo, el archivo `vehículos.csv` tuviese 20 marcas distintas de vehículos, esta función debería crear dinámicamente 20 archivos, uno por cada marca y cada archivo con sus respectivos modelos.

Ejemplo archivo vehículos.csv

```
Marca,Modelo
Abarth,Grande Punto
Abarth,Punto Evo
Alfa Romeo,164
Alfa Romeo,145
Alfa Romeo,147
Audi,A4
Audi,A6
Audi,S6
Austin,Mini
Austin,Montego
Austin,Maestro
Bentley,Continental
Bentley,Azure
Bmw,Z4
Bmw,Z8
Bmw,X5
Toyota,4 Runner
```

Ejemplo archivo Audi.csv

```
Modelo
A4
A6
S6
```

Ejemplo archivo Bentley.csv

```
Modelo
Continental
Azure
```

Ejemplo archivo Toyota.csv

```
Modelo
4 Runner
```

[illegible]

This image shows a full page of blank graph paper. The grid consists of small, equal-sized squares formed by thin black lines. There are 20 columns and 20 rows of squares, creating a total of 400 square units. The paper is otherwise completely blank, with no margins, text, or other markings.

[illegible]

Hoja de Respuesta

This image shows a full page of blank graph paper. The grid consists of small, equal-sized squares formed by thin black lines. There are 20 columns and 20 rows of squares, creating a total of 400 square units. The grid covers the entire area of the page, leaving no margins or other markings.

[illegible]

Hoja de Respuesta

[illegible]