

PROGRAMACIÓN ESTRUCTURADA - O2024

Evaluación Asíncrona 2

Fecha de entrega: 28 de octubre 2024, 11:59pm

Vectores y Matrices

Como parte de su formación como ingenieros, deben cursar en algún punto de su vida (si no es que actualmente) la materia de Álgebra Lineal. El álgebra lineal tiene un sinnúmero de aplicaciones en la resolución matemática de problemas cotidianos; una de las principales herramientas que se utilizan son los vectores y las matrices (vectores de vectores). Podemos visualizarlos de la siguiente manera:

V ₁
V ₂
V ₃
V ₄
V ₅
V ₆

Vector: $n \times 1$

m _{1,1}	m _{1,2}	m _{1,3}	m _{1,4}	m _{1,5}
m _{2,1}	m _{2,2}	m _{2,3}	m _{2,4}	m _{2,5}
m _{3,1}	m _{3,2}	m _{3,3}	m _{3,4}	m _{3,5}
m _{4,1}	m _{4,2}	m _{4,3}	m _{4,4}	m _{4,5}
m _{5,1}	m _{5,2}	m _{5,3}	m _{5,4}	m _{5,5}
m _{6,1}	m _{6,2}	m _{6,3}	m _{6,4}	m _{6,5}

Matriz: $n \times m$

Podemos definir la dimensión de los vectores como n , donde n es el número de elementos, y su equivalente sería $n \times 1$, indicando que tiene n filas y 1 columna. De manera similar, las matrices se definen de dimensión $n \times m$, donde n es el número de filas y m el número de columnas. De esta forma podemos ver que el vector que se muestra arriba es de tamaño o dimensión 6 y que la matriz es de 6×5 .

Para identificar cada uno de los elementos que conforman la matriz o el vector, utilizamos subíndices para obtener la coordenada correspondiente. Cuando hablamos de un vector, simplemente basta con poner como subíndice el número de elemento al que nos referimos, que es el equivalente al número de fila que ocupa. Por otro lado, en las matrices utilizamos el primer número para indicar la fila y el segundo para indicar la columna; por lo que el subíndice $m_{6,4}$ hace referencia al elemento en la fila 6, columna 4.

Parte I

Implementa y utiliza tu propia librería de matrices con la que puedas crear, llenar y operar matrices de la siguiente manera:

- Crear matrices de $N \times N$ (debes llenar tu matriz al momento de crearla)
- Sumar/restar dos matrices
- Obtener la transpuesta de una matriz
- Multiplicar dos matrices
- Imprimir una matriz

Deberás utilizar tu librería desde el archivo main.c en donde se crearán tres matrices (A, B, C) con las cuales vamos a realizar las operaciones.

Parte II

Las funciones deberán definirse y recibir e implementar lo siguiente

- **create_matrix:** recibe una matriz de dimensión NxN y recorre cada casilla preguntando el valor para cada una. Al final imprime la matriz de manera grafica y devuelve la matriz resultante.
- **print_matrix:** recibe una matriz y la imprime de manera gráfica.
- **sum_matrix:** recibe dos matrices y calcula la matriz resultante de la suma de ambas.
- **substrat_matrix:** igual a sum_matrix pero calcula una matriz con la resta de la primera matriz que recibe menos la segunda.
- **multiply_matrix:** recibe dos matrices y calcula la matriz resultante de la multiplicación de estas.
- **transpose_matrix:** recibe una matriz y obtiene su transpuesta

0	1	2
3	-4	5
6	7	-8

A = 3x3

0	3	6
1	4	7
2	5	8

B = 3x3

5	14	25
8	18	30
-9	6	21

(A*B) = 3x3

Parte III

Una vez que tengas tu librería, la utilizarás en tu programa principal.

- Crea al menos tres matrices globales A, B y C (tamaño NxN, libre hasta de 10x10) y llénalas usando la función **create_matrix**.
- Crea matrices globales donde en cada una guardes el resultado de las operaciones de suma, resta y multiplicación dos de las matrices originales (A, B, C). Para este procedimiento utiliza las funciones **sum_matrix**, **subtract_matrix**, **multiply_matrix**.
- Crea otra matriz global y guarda en ella la suma de dos de las matrices originales (A, B, C) e imprime las matrices operando y la matriz resultante. Utiliza tu función **sum_matrix**.
- Crea una matriz global y guarda en ella la resta dos de las matrices originales (A, B, C) e imprime las matrices operando y la matriz resultante. Utiliza tu función **subtract_matrix**.
- Crea una matriz global y guarda en ella la multiplicación dos de las matrices originales (A, B, C) e imprime las matrices operando y la matriz resultante. Utiliza tu función **multiply_matrix**.
- Obtén la matriz transpuesta de la matriz obtenida de la multiplicación. Imprime la original y la resultante.

Ejemplo de ejecución

```
>> Matriz A
>> Ingresa el valor 0,0: 0
>> Ingresa el valor 0,1: 1
>> Ingresa el valor 0,2: 2
>> Ingresa el valor 1,0: 3
...
| 0 | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |
>> Matriz B
>> Ingresa el valor 0,0: 0
...
>> Matriz C
>> Ingresa el valor 0,0: 0
...
>> Suma de matrices A y B
Matriz A:
| 0 | 3 | 6 |
| 1 | 4 | 7 |
| 2 | 5 | 8 |
Matriz B:
| 0 | 1 | 2 |
| 3 | 4 | 5 |
```

```
| 6 | 7 | 8 |  
Resultado:  
| 0 | 4 | 8 |  
| 4 | 0 | 12 |  
| 8 | 12 | 0 |  
>> Resta de matrices B y C  
...  
>> Multiplicacion de matrices A y C  
...  
Resultado:  
| 5 | 14 | 25 |  
| 8 | 18 | 30 |  
| -9 | 6 | 21 |  
>> Matriz A*C transpuesta  
Original:  
| 5 | 14 | 25 |  
| 8 | 18 | 30 |  
| -9 | 6 | 21 |  
Transpuesta  
| 5 | 8 | -9 |  
| 14 | 18 | 6 |  
| 25 | 30 | 21 |
```

Criterios de Evaluación:

Entregables

1. Documento PDF que contiene:
 - a. Portada: logotipo, materia, profesor, título del proyecto, integrantes del equipo
 - b. Introducción. Cuál es la motivación por desarrollar la aplicación, desde el punto de vista didáctico y práctico. Didáctico: qué habilidades se pretenden poner en evidencia con este proyecto. Práctico: para qué le servirá al usuario final la aplicación desarrollada.
 - c.
 - d. Pantallas de la ejecución de la aplicación, acompañada cada una con una explicación.
2. Código fuente en C de la aplicación y librerías desarrolladas
3. Ejecutable (.exe)

Ponderación



Reporte PDF 3 pts

Código Fuente en C 7 pts

Ejecutable 5 pts

Total 15 pts

1. Reporte PDF:
 - a. Para que el reporte tenga validez, deberá justificar lo que se está presentando en el código C; es decir, define la problemática a resolver y explica cuál fue su aproximación para resolverlo. Preferentemente utilizar diagramas que muestren gráficamente la solución propuesta o la base para esta.
2. Código en C
 - a. Debe concordar con la aproximación o solución descrita en el reporte PDF.
 - b. Debe estar ordenado y ser fácil de seguir
3. Tener segmentado todo el programa por funciones, que reciban los parámetros necesarios.

Cada punto se calificará en un rango de puntos dependiendo de la funcionalidad, así como el conocimiento de esta:

- | | |
|---|-------------|
| • Creación y utilización correcta de la librería creada (Parte 1) | [0 - 10]pts |
| • implementación correcta de las funciones (Parte 2) | [0 - 30]pts |
| • Uso adecuado y lógico de las funciones (Parte 3) | [0 - 25]pts |
| • Uso correcto de funciones, paso de parámetros y ciclos | [0 - 25]pts |
| • Estética del código (identación, etc) | [0 - 5] pts |
| • Documentación y comentarios en código | [0 - 5]pts |
| • Si el código no compila | [-100]pts |

Equipos: