

TALLER GRUPAL

Estudiantes: Juan Pablo Landi y Flor Valdivieso

EJERCICIO 01

-----ANALISIS-----

- Variables a utilizar
 - * Limite de notas, limite de estudiantes, promedio, suma del promedio, nota máxima, nota mínima, Arreglo de los promedios, Matriz para las notas.

DATOS DE ENTRADA	PROCESO	SALIDA
(Todos los datos se deben incorporar en los sistemas así que no hay datos que el usuario coloque)	Generamos notas aleatorias para 28 estudiantes en el rango de 0 a 9 y las almacenamos en la matriz. Calculamos el promedio para cada estudiante y el promedio general multiplicando las notas por 0.35 , 0.35 , 0.30 para sumar después el resultado y así obtener el promedio de cada estudiante	Tabla con los nombres de los estudiantes, sus notas en tres categorías, y su promedio y también el promedio general Muestra los estudiantes que tienen notas por encima y por debajo del promedio general. Encuentra y muestra al estudiante con la nota más alta y más baja.

-----PSEUDOCODIGO-----

Algoritmo ejercicio_1

Definir limEst, limNotas como Entero

Definir promedio, sumaPro, notaMax, notaMin como Real

Definir matNotas como Matriz de Real

Definir arrProm como Arreglo de Real

limEst = 28

limNotas = 3

matriz matNotas[limEst][limNotas]

arreglo arrProm[limEst]

promedio = 0

sumaPro = 0

Para (i = 0; i < limEst; i+1) hacer

Para (int j = 0; j < limNotas; j+1) hacer

matNotas[i][j] = GenerarNumeroAleatorio(0, 9)

Fin Para

Fin Para

Para (i = 0; i < limEst; i+1) hacer

arrProm[i] = (matNotas[i][0] * 0.35) + (matNotas[i][1] * 0.35) + (matNotas[i][2] * 0.30)

sumaPro = sumaPro + arrProm[i]

promedio = sumaPro / limEst

Fin Para

Escribir "Nombres | ACD | APE | AA | Promedio"

Para t i = 0; i < limEst; i+1) hacer

Escribir "Estudiante ", i

Para (j = 0; j < limNotas; j+1) hacer

Escribir "| ", matNotas[i][j]

Fin Para

Escribir "| ", arrProm[i]

Salto de línea

Fin Para

Escribir "Promedio: "

Escribir promedio

Salto de línea

Escribir "Estudiantes encima del promedio"

Salto de línea

Para (i = 0; i < limEst; i+1) hacer

Si arrProm[i] > promedio entonces

 Escribir "Estudiante ", i

 Escribir " | ", arrProm[i]

 Salto de línea

Fin Si

Fin Para

Escribir Salto de linea

Escribir "Estudiantes debajo del promedio"

Salto de línea

Para (i = 0; i < limEst; i+1) hacer

 Si arrProm[i] < promedio entonces

 Escribir "Estudiante ", i

 Escribir " | ", arrProm[i]

 Salto de línea

 Fin Si

Fin Para

notaMax = arrProm[0]

notaMin = arrProm[0]

Para (i = 0; i < limEst; i+1) hacer

 Si arrProm[i] > notaMax entonces

 notaMax = arrProm[i]

 Fin Si

 Si arrProm[i] < notaMin entonces

 notaMin = arrProm[i]

Fin Si

Fin Para

Escribir Salto de linea

Escribir "Estudiante con la nota Mayor: "

Escribir notaMax

Escribir "Estudiante con la nota Menor: "

Escribir notaMin

Fin Algoritmo

-----PRUEBA DE ESCRITORIO-----

Nombres ACD APE AA Promedio	Promedio:	Estudiantes encima del promedio	Estudiantes debajo del promedio	Estudiante con la nota Mayor:	Estudiante con la nota Menor:
Estudiante 0 7.02 2.89 8.47 6.43 Estudiante 1 4.46 5.35 1.01 3.79 Estudiante 2 1.66 2.37 5.18 3.22 Estudiante 3 3.82 5.94 4.32 4.93 Estudiante 4 0.69 7.46 7.92 5.63 Estudiante 5 4.67 3.44 2.16 3.60 Estudiante 6 8.12 0.99 9.68 6.58 Estudiante 7 0.64 8.77 0.09 3.33	5.18	Estudiante 0 6.43 Estudiante 4 5.63 Estudiante 6 6.58 Estudiante 8 9.88 Estudiante 10 5.79 Estudiante 17 8.26 Estudiante 18 7.52 Estudiante 20 5.24 Estudiante 21 5.76 Estudiante 23 7.61 Estudiante 24 5.45 Estudiante 26 5.84 Estudiante 27 6.17	Estudiante 1 3.79 Estudiante 2 3.22 Estudiante 3 4.93 Estudiante 5 3.60 Estudiante 7 3.33 Estudiante 9 4.41 Estudiante 11 3.69 Estudiante 12 2.75 Estudiante 13 4.31 Estudiante 14 3.88 Estudiante 15 4.44 Estudiante 16 4.76 Estudiante 19 1.97 Estudiante 22 4.90 Estudiante 25 5.03	9.88	1.97

Estudiante 8	9.99				
	9.41				
	8.83				
	9.88				
Estudiante 9	8.99				
	2.60				
	1.01				
	4.41				
Estudiante 10					
	2.57				
	8.09				
	5.89				
	5.79				
Estudiante 11					
	1.69				
	1.96				
	6.90				
	3.69				
Estudiante 12					
	2.24				
	0.45				
	5.15				
	2.75				
Estudiante 13					
	5.30				
	4.88				
	2.12				
	4.31				
Estudiante 14					
	2.83				
	6.44				
	1.81				
	3.88				
Estudiante 15					
	9.38				
	1.98				
	1.33				
	4.44				
Estudiante 16					
	6.00				
	5.96				
	1.63				
	4.76				
Estudiante 17					
	8.39				
	9.15				
	6.06				
	8.26				
Estudiante 18					
	8.66				
	6.05				
	6.77				
	7.52				
Estudiante 19					
	3.03				
	2.23				
	0.37				
	1.97				
Estudiante 20					
	6.51				
	3.84				
	4.64				

	5.24				
Estudiante 21	4.78				
	7.38				
	4.31				
	5.76				
Estudiante 22	2.55				
	4.07				
	7.39				
	4.90				
Estudiante 23	4.44				
	7.30				
	10.00				
	7.61				
Estudiante 24	8.71				
	2.06				
	4.81				
	5.45				
Estudiante 25	1.85				
	4.55				
	7.97				
	5.03				
Estudiante 26	6.10				
	1.05				
	9.53				
	5.84				
Estudiante 27	5.52				
	6.67				
	5.45				
	6.17				

EJERCICIO 2

ANALISIS

DATOS DE ENTRADA	PROCESO	SALIDA
Nombre	Nombre del producto y verificar	Catalogo
Precio	si hay disponible	
Cantidad	Precio y Cantidad	

PSEUDOCODIGO

Algoritmo ejercicio_2

Definir nomProducto (6) como cadena [Gaseosas, picaditas, alcohol, aliños, granos secos, embutidos]

Definir limProducto =6, columnas=2, productoIng como enteros

Definir matriz llamada matPreCan (6)(2) como real [limProductos][columnas]

Para (i=0; i<limProductos;i+1)

 Escribir “Aquí ingrese los precios señor emprendedor”

 matPreCan [0][0] = 1.00

 matPreCan [1][0] = 0.50

 matPreCan [2][0] = 2.50

 matPreCan [3][0] = 0.50

 matPreCan [4][0] = 0.75

 matPreCan [5][0] = 1.50

Para (j=0; j<columnas;j+1)

 Escribir “Aquí ingrese la cantidad señor emprendedor”

 matPreCan [0][1] = 100

 matPreCan [1][1] = 10

 matPreCan [2][1] = 50

 matPreCan [3][1] = 50

 matPreCan [4][1] = 75

 matPreCan [5][1] = 80

Fin Para

Fin Para

Escribir “¿Qué producto busca?”

Leer nomProducto

Si (productoIng == nomProducto)

 Escribir “Si tenemos ese producto, aquí tiene nuestro catalogo”

Sino

 Escribir “Producto no disponible”

FinSi

Para (i=0; i<limProductos;i+1)

 nomProdcuto[i]

 Para (j=0; j<columnas;j+1)

 matPreCan[i][j]

Fin Para

Fin Para

Fin Algortimo

-----PRUEBA DE ESCRITORIO-----

Producto	Respuesta	Catalogo																		
Gaseosa	Si tenemos ese producto, aquí nuestro catalogo.	<table><tr><td>Gaseosa</td><td>1.00</td><td>100</td></tr><tr><td>Picaditas</td><td>0.50</td><td>10</td></tr><tr><td>Alcohol</td><td>2.50</td><td>50</td></tr><tr><td>Aliños</td><td>0.50</td><td>50</td></tr><tr><td>Granos</td><td>0.75</td><td>75</td></tr><tr><td>Embutidos</td><td>1.50</td><td>80</td></tr></table>	Gaseosa	1.00	100	Picaditas	0.50	10	Alcohol	2.50	50	Aliños	0.50	50	Granos	0.75	75	Embutidos	1.50	80
Gaseosa	1.00	100																		
Picaditas	0.50	10																		
Alcohol	2.50	50																		
Aliños	0.50	50																		
Granos	0.75	75																		
Embutidos	1.50	80																		

EJERCICIO 03

-----ANALISIS-----

- Para los datos de la tienda o los productos tenemos que incorporarlos directamenet al sistema y en el programa lo hicimos asi
- 01 Gaseosas 1.0 100
- 002 Picaditas 1.0 100
- 003 Alcohol 2.5 100

ENTRADA	PROCESO	SALIDA
Codigo Cantidad	Comparación si el producto existe Comparamos si la cantidad es menor a la existente Calculamos el total multiplicando la cantidad por el precio unitario Calculamos el iva multiplicando el total por 1.12 Comprobamos si hay algún descuento solo si la compra supera la cantidad de 100 , multiplicando por 0.1	Nombre del producto Cantidad Precio por unidad Total Descuento Total con el iva

-----PSEUDOCODIGO-----

Algoritmo ejercicio_3

Definir total, totalConIVA, descuento como Real

Definir matriz llamada "inventario" como Cadena

Definir codigoProducto como Cadena

Definir cantidadDeseada como Entero

Definir productoEncontrado como Booleano

total = 0

totalConIVA = 0

descuento = 0

inventario = (

 ["001", "Gaseosas", "1.0", "100"],

 ["002", "Picaditas", "1.0", "100"],

 ["003", "Alcohol", "2.5", "100"]

)

Escribir "CÓDIGO | PRODUCTO | PRECIO | CANTIDAD"

Para (Cadena[] producto : inventario (para que sirve para recorrer los productos ya dados)) hacer

 Escribir producto[0], producto[1], producto[2], producto[3]

Fin Para

Escribir "Ingrese el código del producto:"

Leer codigoProducto

Escribir "Ingrese la cantidad deseada:"

Leer cantidadDeseada

productoEncontrado = Falso

Si producto[0] igual a codigoProducto entonces

 productoEncontrado = Verdadero

 precioUnitario = ConvertirADecimal(producto[2])

 cantidadDisponible = ConvertirAEntero(producto[3])

Si cantidadDeseada > cantidadDisponible entonces

 Escribir "Disculpe, se ha agotado este producto"

Fin Si

$total = total + (cantidadDeseada * precioUnitario)$

Si $total > 100.0$ entonces

$descuento = total * 0.1$

Fin Si

$totalConIVA = total * 1.12$

Fin Si

Fin Para

Si no(expresión booleana) productoEncontrado entonces

Escribir "Disculpe, no poseemos este producto"

Sino

Escribir "FACTURA"

Escribir "Producto: ", inventario[0][1]

Escribir "Cantidad: ", cantidadDeseada

Escribir "Precio por unidades: \$", inventario[0][2]

Escribir "Total: \$", total

Escribir "Descuento: \$", descuento

Escribir "Total Con IVA: \$", totalConIVA

Fin Si

Fin Algoritmo

-----PRUEBA DE ESCRITORIO-----

ENTRADA CODIGO	CANTIDAD DESEADA	Buscar según el código el	TOTAL	Descuento	Total con Iva	FACTURA
001	10	Producto existente	$0 + (10 * 1) = 10$	$10 > 100$ (no) No hay descuento	$10 * 1.12 = 11.20$	Producto: Gaseosas Cantidad: 10 Precio por unidades: \$1.0 Total: \$10.0 Descuento: \$0.0 Total Con IVA: <u>\$11.2000000000000001</u>

EJERCICIO 04

ANALISIS

- Para la prueba este programa tendremos que hacer un análisis básico de que nomas tenemos que colocar
- Antes de todo definimos el tablero en el sistema
- `tablero = [[' ', ' ', ' '], [' ', ' ', ' '], [' ', ' ', ' ']]`

ENTRADA	PROCESO	SALIDA
Coordenadas de cada posición en el tablero	Comprobación de todo	El tablero de cada posición puesta

PSEUDOCODIGO

Algoritmo juego_3_en_raya

Definir tablero como Matriz de Caracteres

Definir juegoEnCurso como Booleano

Definir jugadorActual como Caracter

`tablero = [[' ', ' ', ' '], [' ', ' ', ' '], [' ', ' ', ' ']]`

`juegoEnCurso = Verdadero`

`jugadorActual = 'X'`

Escribir "Bienvenidos al juego de 3 en raya: Diviertete ñaño"

Mientras juegoEnCurso hacer

Escribir "-----"

Para (i = 0; i < 3; i+1) hacer

Escribir "| "

Para (j = 0; j < 3; j+1) hacer

Escribir `tablero[i][j], " | "`

Fin Para

Escribir ""(para hacer un salto de linea)

Escribir "-----"

Fin Para

Hacer {

Escribir "Jugador ", jugadorActual, ", ingrese fila (0-2) y columna (0-2) separadas por espacio:"

Leer fila, columna

} Mientras (fila < 0 o fila >= 3 o columna < 0 o columna >= 3 o tablero[fila][columna] diferente de ' ')

tablero[fila][columna] =jugadorActual

Para (i = 0; i < 3; i+1) hacer

Si (tablero[i][0] igual a jugadorActual y tablero[i][1] igual a jugadorActual y tablero[i][2] igual a

jugadorActual) o (tablero[0][i] igual a jugadorActual y tablero[1][i] igual a jugadorActual
tablero[2][i] igual a jugadorActual) entonces

juegoEnCurso = Falso

Detener

Fin Si

Fin Para

Si (tablero[0][0] igual a jugadorActual y tablero[1][1] igual a jugadorActual y tablero[2][2] igual a
jugadorActual) o (tablero[0][2] igual a jugadorActual y tablero[1][1] igual a jugadorActual y tablero[2][0]
igual a jugadorActual) entonces

juegoEnCurso = Falso

Escribir "Alguien ganó, que bien ñaño"

Fin Si

empate = Verdadero

Para (i = 0; i < 3; i++) hacer

Para (j = 0; j < 3j++) hacer

Si tablero[i][j] igual a ' ' entonces

empate = Falso

Detener

Fin Si

Fin Para

Si no empate entonces

Detener

Fin Si

Fin Para

Si empate entonces

juegoEnCurso = Falso

Escribir "Nadie ha ganado, jueguen una más"

Fin Si

Si jugadorActual igual a 'X' entonces

jugadorActual = 'O'

Sino

jugadorActual = 'X'

Fin Si

Fin Mientras

Fin Algoritmo

-----PRUEBA DE ESCRITORIO-----

ENTRADA	PROCESO	SALIDA
2,1	Verficiacion de las coordenadas correctas Comprobación de el jugador que esta turnando Colocación de la letra según el jugador	----- ----- ----- X -----
1 ,1	Verficiacion de las coordenadas correctas Comprobación de el jugador que esta turnando Colocación de la letra según el jugador	----- ----- O ----- X -----
2,0	Verficiacion de las coordenadas correctas Comprobación de el jugador que esta turnando	----- ----- O -----

	Colocación de la letra según el jugador	----- X X
1, 1	Verficiacion de las coordenadas correctas (coordenada ya ocupada, volvemos al ciclo repetitivo) Comprobación de el jugador que esta turnando Colocación de la letra según el jugador	N/A
0,0	Verficiacion de las coordenadas correctas Comprobación de el jugador que esta turnando Colocación de la letra según el jugador	----- O ----- O ----- X X -----
2,2	Verficiacion de las coordenadas correctas Comprobación de el jugador que esta turnando Colocación de la letra según el jugador	Alguien ganó, que bien ñaño

EJERCICIO 05

-----ANALISIS-----

- Para la multiplicación de las matrices es colocar ambas matrices son sus miembros y dar los números a cada una de ellas
- Para esto hacemos una generación automática de cada numero
- Por lo tanto no habrá datos de entrada

ENTRADA	PROCESO	SALIDA
n/a	Generacion de cada numero en un rango de -9 ,9 Multipliacion de ambas matrices y mostrar el resultado como una matriz tercera	Mostrar la matriz de respuesta

-----PSEUDOCODIGO-----

Algoritmo multiplicarMatrices

Definir limMatriz como Entero

Definir primeraMat como Matriz de Enteros

Definir segundaMat como Matriz de Enteros

Definir solucion como Matriz de Enteros

limMatriz =3

primeraMat[limMatriz][limMatriz]

segundaMat[limMatriz][limMatriz]

solucion[limMatriz][limMatriz]

Para (i = 0; i < limMatriz; i+1)hacer

Para (j = 0; j < limMatriz; j+1)hacer

primeraMat[i][j] = GenerarNumeroAleatorio(-9, 9)

Fin Para

Fin Para

Escribir "Primera Matriz: "

Para (i = 0; i < primeraMat; i+1)hacer

Para (j = 0; j < limMatriz; j+1)hacer

Escribir primeraMat[i][j]

Fin Para

Escribir ""

Fin Para

Para (i = 0; i < limMatriz; i+1)hacer

Para (j = 0; j < limMatriz; j+1)hacer

segundaMat[i][j] = GenerarNumeroAleatorio(-9, 9)

Fin Para

Fin Para

Escribir "Segunda Matriz: "

Para (i = 0; i < segundaMat ; i+1)hacer

Para (j = 0; j < limMatriz; j+1)hacer

```

        Escribir segundaMat[i][j],
    Fin Para
    Escribir ""
Fin Para
Para ( i = 0; i < limMatriz; i+1) hacer
    Para ( j = 0; j < limMatriz; j+)hacer
        Para ( k = 0; k < limMatriz; k+1)hacer
            solucion[i][j] = 0
            solucion[i][j] = solucion[i][j] + primeraMat[i][k] * segundaMat[k][j]
        Fin Para
    Fin Para
Fin Para
Escribir "Resultado: "
Para (i = 0; i < limMatriz; i+1)hacer
    Para ( j = 0; j < limMatriz; j+1)hacer
        Escribir solucion[i][j]
    Fin Para
Escribir ""
Fin Para
Fin Algoritmo

```

-----PRUEBA DE ESCRITORIO-----

ENTRADA	GENERACION DE LA PRIMERA MATRIZ			GENERACION DE LA SEGUNDA MATRIZ			MULTIPLICACION	SALIDA
N/A	-5	-6	8	-4	5	1	primeraMat[i][k] * segundaMat[k][j]	-72 29 39
	-6	7	7	6	-5	-2		17 -44 8
	-3	-1	-3	-7	3	4		27 -19 -13

EJERCICIO 06

ANALISIS

- Debemos generar un limite de las matrices cuadradas
- Una vez generado esto debemos generar números aleatorios entre -9,9

INICIO	INTERMEDIO	FIN
El limite de la matriz ya esta colocado Generamos las matrices con números -9,9	Sumar o restar ambas matrices según el usuario lo pida	Presentar el resultado de la matriz ya sea sumada o restada

PSEUDOCODIGO

Algoritmo operacionesMatrices

Definir limMatriz como Entero

Definir primeraMat como Matriz de Enteros

Definir segundaMat como Matriz de Enteros

Definir solucion como Matriz de Enteros

Definir opcion como Entero

limMatriz = 3

primeraMat[limMatriz][limMatriz]

segundaMat[limMatriz][limMatriz]

solucion[limMatriz][limMatriz]

Para (i = 0; i < limMatriz; i+1)hacer

Para (j = 0; j < limMatriz; j+1)hacer

primeraMat[i][j] = GenerarNumeroAleatorio(-9, 9)

Fin Para

Fin Para

Escribir "Primera Matriz: "

```

Para (i = 0; i < primeraMat; i+1)hacer
    Para (j = 0; j < limMatriz; j+1)hacer
        Escribir primeraMat[i][j]
    Fin Para
    Escribir ""
Fin Para

Para (i = 0; i < limMatriz; i+1)hacer
    Para ( j = 0; j < limMatriz; j+1)hacer
        segundaMat[i][j] = GenerarNumeroAleatorio(-9, 9)
    Fin Para
Fin Para

Escribir "Segunda Matriz: "

Para ( i = 0; i < segundaMat; i+1)hacer
    Para (j = 0; j < limMatriz; j+1)hacer
        Escribir segundaMat[i][j]
    Fin Para
    Escribir ""
Fin Para

Escribir "Si desea sumar las matrices ingrese 1"
Escribir "Si desea restar las matrices ingrese 2"

Leer opcion

Si opcion ==1 Entonces
    Escribir "Resultado"
    Para ( i = 0; i < solucion; i+1)hacer
        Para ( j = 0; j < solucion; j+1) hacer
            solucion[i][j] = primeraMat[i][j] + segundaMat[i][j]
            Escribir solucion[i][j]
        Fin Para
    Fin Para
    Escribir ""

```

```

Fin Para
Sino Si opción == 2 Entonces
    Escribir "Resultado: "
    Para ( i = 0; i < solucion; i+1)hacer
        Para (int j = 0; j < solucion; j+1)hacer
            solucion[i][j] <- primeraMat[i][j] - segundaMat[i][j]
            Escribir solucion[i][j]
        Fin Para
    Fin Para
    Escribir ""
Fin Para
Fin si
Fin Algoritmo

```

-----PRUEBA DE ESCRITORIO-----

Generación de ambas matrices	Numero de la operación ingresado	Realización de la operación correspondiente	Impresión de la solución		
Primera Matriz: -1 2 0 7 -2 8 3 -6 8 Segunda Matriz: -4 -2 4 -3 7 0 2 7 -7		Resta de las matrices primeraMat[i][j]-segundaMat[i][j]	3	4	-4
			10	-9	8
			1	-13	15
Primera Matriz: 1 -6 6 5 6 8	1	Suma de ambas matrices: primeraMat[i][j]+segundaMat[i][j]	9	-3	7
			6	10	8
			-4	-11	-6

-6	-7	-5			
Segunda Matriz:					
8	3	1			
1	4	0			
2	-4	-1			

EJERCICIO 07

ANALISIS

- No habrá ningún dato de entrada porque todo lo genera automáticamente el programa
- Tenemos que hacer se muestre la diagonal secundaria que es la contraria a la diagonal principal

ENTRADA	PROCESO	SALIDA
n/a	<p>Se utiliza un bucle anidado para imprimir cada elemento de la matriz en filas y columnas</p> <p>Se utiliza un bucle para recorrer la matriz en orden ascendente en las filas y descendente en las columnas, imprimiendo los elementos de la diagonal secundaria.</p> <p>Se utiliza un bucle anidado para recorrer cada fila y cada columna</p> <p>Se imprimen los elementos que están por encima de la diagonal secundaria</p> <p>Se utiliza un bucle anidado para recorrer cada fila y cada columna</p> <p>Se imprimen los elementos que están debajo de la diagonal secundaria</p>	<p>La diagonal secundaria (invertida)</p> <p>Los elementos que están arriba de esta y los q están debajo de esta.</p>

PSEUDOCODIGO

Algoritmo operacionesMatriz

Definir limMat como Entero

Definir mat como Matriz de Enteros

limMat = 3

matriz mat[limMat][limMat]

Para (i = 0; i < limMat; i+1)hacer

Para (j = 0; j < limMat; j+1)hacer

mat[i][j] = GenerarNumeroAleatorio(-9, 9)

Fin Para

Fin Para

Para (i = 0; i < mat; i+1) hacer

Para (j = 0; j < limMat; j+1)hacer

Escribir mat[i][j]

Fin Para

Escribir ""

Fin Para

Para (int i = 0; i < mat; i+1)hacer

Para (int j = 0; j < mat; j+1) hacer

Si i == j

Fin Para

Fin Para

Escribir "Diagonal Secundaria: "

j = (limMat) - 1

Para (i = 0; i < mat; i++)hacer

Escribir mat[i][j--]

Fin Para

Escribir "Elementos encima de la diagonal Secundaria"

Para (i = 0; i < mat; i+1)hacer

Para (k = 0; k < mat ; k+1)hacer

```

        Escribir mat[i][k]

    Fin Para

    Escribir ""

Fin Para

Escribir "Elementos debajo de la diagonal Secundaria "

Para (i = 1; i < mat; i+1)hacer

    Para (k = mat - i; k < mat; k+1)hacer

        Escribir mat[i][k]

    Fin Para

    Escribir ""

Fin Para

Fin Algoritmo

```

-----PRUEBA DE ESCRITORIO-----

Generar elementos de la matriz con limite de 3	Comprobar la diagonal secundaria	Elementos encima	Elemto debajo
<div> <div>0</div> <div>-5</div> <div>-5</div> </div> <div> <div>8</div> <div>5</div> <div>-2</div> </div> <div> <div>-2</div> <div>6</div> <div>1</div> </div>	<p>los elementos de la diagonal secundaria de la matriz van en orden ascendente de las filas y van en descendente de las columnas, lo que es consistente con la definición de la diagonal secundaria en una matriz cuadrada.</p> <p>Y aquí el resultado</p> <div> <div>-5</div> <div>5</div> <div>-2</div> </div>	<p>Utilizamos dos bucles para recorrer y verificar los elementos por encima de la diagonal secundaria, y se imprime cada elemento correspondiente a esa condición. La variable k en el bucle interior se encarga de ajustar la posición en la fila para que solo se consideren las columnas por encima de la diagonal secundaria.</p> <p>Aquí la solución</p> <div> <div>0</div> <div>-5</div> </div>	<p>Utilizamos dos bucles para recorrer y verificar los elementos por debajo de la diagonal secundaria, y se imprime cada elemento correspondiente a esa condición. La variable k en el bucle interior se encarga de ajustar la posición en la fila para que solo se consideren las columnas por debajo de la diagonal secundaria.</p> <p>Aquí la solución</p> <div> <div>-2</div> </div>

		8	6	1
--	--	---	---	---

EJERCICIO 08

-----ANALISIS-----

- No habrá ningún dato de entrada porque todo lo genera automáticamente el programa
- Tenemos que hacer que mmuestre la diagonal principal la cual es una representación de las coodernadas repetidas ejemplo (0,0)(1,1,)(2,2)

-----PSEUDOCODIGO-----

Algoritmo operacionesMatriz

Definir limMat como Entero

Definir mat como Matriz de Enteros

limMat = 3

matriz mat[limMat][limMat]

Para (i = 0; i < limMat; i+1)hacer

Para (j = 0; j < limMat; j+1) hacer

mat[i][j] <- GenerarNumeroAleatorio(-9, 9)

Fin Para

Fin Para

Para (i = 0; i < mat; i+1)hacer

Para (j = 0; j < limMat; j+1)hacer

Escribir mat[i][j]

Fin Para

Escribir ""

Fin Para

Para (i = 0; i < mat; i+1)hacer

Para (j = 0; j < mat; j+1)hacer

Si i == j

```

    Fin Para
Fin Para
Escribir "Diagonal Principal: "
Para ( i = 0; i < limMat; i+1)hacer
    Escribir mat[i][i]
Fin Para
Escribir "Elementos debajo de la diagonal Principal"
Para ( i = 0; i < mat; i+1)hacer
    Para ( j = 0; j < i; j+1)hacer
        Escribir mat[i][j]
    Fin Para
    Escribir ""
Fin Para
Escribir "Elementos encima de la diagonal Principal"
Para ( j = 0; j < mat; j+1)hacer
    Para ( i = 0; i < j; i+1)hacer
        Escribir mat[i][j]
    Fin Para
Fin Para
Fin Algoritmo

```

-----PRUEBA DE ESCRITORIO-----

Generar elementos de la matriz con limite de 3	Comprobar la diagonal	Elementos encima	Elemtos debajo
<pre> 2 -1 -6 -5 -3 5 5 7 -1 </pre>	La condición $i == j$ en la instrucción <code>mat[i][j]</code> asegura que solo se acceda a los elementos de la diagonal principal. Los elementos de la diagonal principal se	el bucle PARA doble anidado <code>for i</code> y <code>for j</code> se encarga de recorrer y verificar los elementos por encima de la diagonal principal, y cada elemento	el bucle PARA doble anidado <code>for i</code> y <code>for j</code> se encarga de recorrer y verificar los elementos por debajo de la diagonal principal, y cada elemento

	imprimen uno por uno en una nueva línea. Tal y como ves aqui 2 -3 -1	correspondiente a esa condición se imprime en la misma línea. La condición $i < j$ en el bucle interno asegura que solo se recorran las filas por encima de la diagonal principal para cada columna j. Como lo ves aqui 5 5 7	correspondiente a esa condición se imprime en una nueva línea. La condición $j < i$ en el bucle interno asegura que solo se recorran las columnas por debajo de la diagonal principal para cada fila i Como lo ves aqui -1 -6 5
--	-----------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

EJERCICIO 09

-----ANALISIS-----

- Para la pirámide definir la altura de la pirámide en el programa lo que significa que no habrá datos de entrada

ENTRADA	PROCESO	SALIDA
n/a	El limite esta en el programa Ejecutar las veces según el limite de la pirámide Uno que controle el numero de asteriscos Luego imprimir los asteriscos	Piramide

-

-----PSEUDOCODIGO-----

Algoritmo imprimirPiramide

Definir limPiramide como Entero

limPiramide = 4

Para (i = 0; i < limPiramide; i+1)hacer

Para (j = 0; j < i; j+1)hacer

Escribir "*" "

Fin Para

Escribir ""

Fin Para

Fin Algoritmo

-----PRUEBA DE ESCRITORIO-----

Declaracion de limite	Para externo	Para interno	Impresión de asteriscos y saltos de linea
4	Controla las filas y repite el bucle 4 veces	controla las columnas de cada fila. El límite del bucle interno $j \leq i$ asegura que se impriman asteriscos en una cantidad igual al número de la fila actual i .	Después de imprimir los asteriscos en una fila, se agrega un salto de línea para pasar a la siguiente fila. Este proceso se repite hasta que se haya completado la pirámide con la altura deseada. Al ejecutar el programa, verás una salida similar a la siguiente: <pre>* * * * * * * * * *</pre>

EJERCICIO 10

-----ANALISIS-----

- Para esta pirámide completa se puede hacer una pirámide normal y una invertida justo adebajo para poder completar un rombo

ENTRADA	PROCESO	SALIDA
El limite de la pirámide(base)	Tenemos que crear dos pirámides una acendente y una desendente para cada uno de estos ocupamos 3 para , con las variables i,j,k	La pirámide completa acendente y desendente

-----PSEUDOCODIGO-----

Algoritmo imprimirFigura

Definir tamaño como Entero

Escribir "Ingrese el tamaño de la figura: "

Leer tamaño

Para (i = 0; i <= tamaño; i+1)hacer

Para (j = tamaño-i; j > 0; j-1) hacer

Escribir " "

Fin Para

Para (k = 0; k < i; k+1)hacer

Escribir " *"

Fin Para

Escribir ""

Fin Para

Para (i = 0; i <= tamaño; i+1)hacer

Para (j = 0; j <= i; j+1)hacer

Escribir " "

Fin Para

Para (k = tamaño - i - 1; k > 0; k-1) hacer

Escribir " *"

Fin Para

Escribir ""

Fin Para

Fin Algoritmo

-----PRUEBA DE ESCRITORIO-----

Entrada	Para(i) A	Para(j) A	Para (k) A	Para(i) D	Para(j) D	Para(k) D	Salida
5	para controlar las filas de	imprime espacios en blanco antes de	imprime los asteriscos en orden ascendente.	para controlar las filas de la	imprime espacios en blanco antes de	imprime los asteriscos en orden descendente.	* * * * * * * * * * * * * * *

	la pirámide ascendente.	los asteriscos, creando la forma triangular.		pirámide descendente	los asteriscos, creando la forma triangular invertida.		* * * * * * * * * *
--	----------------------------	----------------------------------------------------------	--	-------------------------	-----------------------------------------------------------------------	--	------------------------------