

Programación



Módulo 2. Arreglos

- Arreglos bidimensionales
- Practica 39 - 44

NRC: 42561

Horario: (Me cambie de horario)

Viernes 7:00 am – 10:55 am

Martes y jueves 9:00 am – 11:00 am

Nombre:

Beleche Mendoza Alondra Jazmín

Código: 218565552

30 / Abril / 2023

Práctica 39

Beleche Alondra

ve\Escritorio\1er Semestre\seminario de progra\paty 39.cpp - [Executing] - Dev-C++ 5.11

ar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda

TDM-GCC 4.9.2 64-bit Release

Sin Nombre1 paty 39.cpp

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 /* Práctica 39: Llenar arreglo
5  P: Beleche Alondra */
6
7 int main(int argc, char *argv[])
8 {
9     int matriz[4][4] = { {22,45,126,36}, {98,7,4,514}, {123,46,47,41}, {12,78,38} };
10    int i, j;
11
12    for (i = 0; i < 4; i++)
13    {
14        for (j = 0; j < 4; j++)
15        {
16            printf("Dame el elemento %d,%d del arreglo bidimensional\n", i, j);
17            scanf("%d", &matriz[i][j]);
18        }
19    }
20
21    for (i = 0; i < 4; i++)
22    {
23        for (j = 0; j < 4; j++)
24        {
25            printf("El elemento %d,%d es %d\n", i, j, matriz[i][j]);
26        }
27    }
28
29    return 0;
30 }
31
```

C:\Users\alond\OneDrive\Escritorio\1er S...

Dame el elemento 3,0 del arreglo bidimensional
5
Dame el elemento 3,1 del arreglo bidimensional
4
Dame el elemento 3,2 del arreglo bidimensional
3
Dame el elemento 3,3 del arreglo bidimensional
2
El elemento 0,0 es 1
El elemento 0,1 es 2
El elemento 0,2 es 3
El elemento 0,3 es 2
El elemento 1,0 es 4
El elemento 1,1 es 2
El elemento 1,2 es 3
El elemento 1,3 es 4
El elemento 2,0 es 2
El elemento 2,1 es 1
El elemento 2,2 es 3
El elemento 2,3 es 4
El elemento 3,0 es 5
El elemento 3,1 es 4
El elemento 3,2 es 3
El elemento 3,3 es 2

Process exited after 13.01 seconds with return va
lue 0
Presione una tecla para continuar . . .

Registros de Compilación Depuración Resultados Cerrar

- Output Filename: C:\Users\alond\OneDrive\Escritorio\1er Semestre\seminario de progra\paty 39.cpp
- Output Size: 128.6015625 KiB
- Compilation Time: 0.31s

Sel: 0 Lines: 31 Length: 660 Insertar Done parsing in 0.141 seconds

Práctica 40

Beleche Alondra

ve\Escritorio\1er Semestre\seminario de progra\p 40.cpp - [Executing] - Dev-C++ 5.11

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 /* Practica 40: Llena arreglos
5  P: Beleche Alondra */
6
7 int main(int argc, char *argv[]) {
8     int A1[3][3], B1[3][3], C1[3][3], i, j;
9
10    for (i = 0; i < 3; i++) {
11        for (j = 0; j < 3; j++) {
12            printf("Dame el elemento %d, %d del arreglo A1\n", i, j);
13            scanf("%d", &A1[i][j]);
14
15            printf("Dame el elemento %d, %d del arreglo B1\n", i, j);
16            scanf("%d", &B1[i][j]);
17        }
18    }
19
20    for (i = 0; i < 3; i++) {
21        for (j = 0; j < 3; j++) {
22            C1[i][j] = A1[i][j] + B1[i][j];
23            printf("%d + %d = %d\n", A1[i][j], B1[i][j], C1[i][j]);
24        }
25    }
26
27    return 0;
28 }
29
```

Output:

```
Dame el elemento 1, 2 del arreglo A1
4
Dame el elemento 1, 2 del arreglo B1
5
Dame el elemento 2, 0 del arreglo A1
6
Dame el elemento 2, 0 del arreglo B1
7
Dame el elemento 2, 1 del arreglo A1
8
Dame el elemento 2, 1 del arreglo B1
3
Dame el elemento 2, 2 del arreglo A1
2
Dame el elemento 2, 2 del arreglo B1
1
1 + 2 = 3
3 + 4 = 7
8 + 9 = 17
3 + 6 = 9
7 + 2 = 9
4 + 5 = 9
6 + 7 = 13
8 + 3 = 11
2 + 1 = 3

-----
Process exited after 17.48 seconds with return value 0
Presione una tecla para continuar . . .
```

Compilation Log:

```
- Output Filename: C:\Users\alond\OneDrive\Escritorio\1er Semestre\seminario de progra\p 40.cpp
- Output Size: 129.1015625 KiB
- Compilation Time: 0.38s
```

Status Bar: Sel: 0 Lines: 29 Length: 729 Insertar Done parsing in 0.031 seconds

Práctica 4I

Beleche Alondra

e\Escritorio\1er Semetre\seminario de progra\p 41.cpp - [Executing] - Dev-C++ 5.11

Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda

TDM-GCC 4.9.2 64-bit Release

paty 39.cpp p 40.cpp p 41.cpp

```
1 #include <stdio.h>
2
3 /* Prueba de escritorio, Practica 41
4 P: Beleche Alondra */
5
6 int main(int argc, char *argv[]) {
7     int x[3] = {67, 15, 3};
8     int y[3] = {0};
9     int i, j;
10
11     for (i = 0; i < 3; i++) {
12         for (j = 0; j < 3; j++) {
13             y[i] += x[j];
14         }
15     }
16
17     for (i = 0; i < 3; i++) {
18         printf("Elemento %d: %d\n", i+1, y[i]);
19     }
20
21     return 0;
22 }
23
```

C:\Users\alond\OneDrive\Escritorio\1er Semetre\semin...
Elemento 1: 85
Elemento 2: 85
Elemento 3: 85

Process exited after 0.0711 seconds with return value 0
Presione una tecla para continuar . . .

Registro de Compilación Depuración Resultados Cerrar

- Output Filename: C:\Users\alond\OneDrive\Escritorio\1er Semetre\seminario de progra\p 41.exe
- Output Size: 127.931640625 KiB
- Compilation Time: 0.41s

Sel: 0 Lines: 23 Length: 411 Insertar Done parsing in 0.016 seconds

Práctica 42

Beleche Alondra

ive\Escritorio\1er Semetre\seminario de progra\p.42.cpp - [Executing] - Dev-C++ 5.11

car Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda

TDM-GCC 4.9.2 64-bit Release

als)

paty 39.cpp p 40.cpp p 41.cpp p.42.cpp

```
1 #include <stdio.h>
2
3 /* Practica 42: Suma de columnas
4  P: Beleche Alondra */
5
6 int main(int argc, char *argv[]) {
7     int matriz[3][3], col[3] = {0, 0, 0}, i, j;
8
9     for (i = 0; i < 3; i++) {
10         for (j = 0; j < 3; j++) {
11             printf("Elemento [%d][%d]: ", i, j);
12             scanf("%d", &matriz[i][j]);
13             col[j] += matriz[i][j];
14         }
15     }
16
17     printf("Matriz ingresada:\n");
18     for (i = 0; i < 3; i++) {
19         for (j = 0; j < 3; j++) {
20             printf("%d\t", matriz[i][j]);
21         }
22         printf("\n");
23     }
24
25     printf("Suma de columnas:");
26     for (j = 0; j < 3; j++) {
27         printf("\t%d", col[j]);
28     }
29     printf("\n");
30
31     return 0;
32 }
33
```

C:\Users\alond\OneDrive\Escritorio\1er Semetre\semina...

```
Elemento [0][0]: 1
Elemento [0][1]: 1
Elemento [0][2]: 2
Elemento [1][0]: 3
Elemento [1][1]: 4
Elemento [1][2]: 2
Elemento [2][0]: 3
Elemento [2][1]: 4
Elemento [2][2]: 5
Matriz ingresada:
1      1      2
3      4      2
3      4      5
Suma de columnas:      7      9      9

-----
Process exited after 6.714 seconds with return value 0
Presione una tecla para continuar . . .
```

recursos Registro de Compilación Depuración Resultados Cerrar

```
- Output Filename: C:\Users\alond\OneDrive\Escritorio\1er Semetre\seminario de progra\p.42.exe
- Output Size: 129.4423828125 KiB
- Compilation Time: 2.00s
```

Sel: 0 Lines: 33 Length: 722 Insertar Done parsing in 0.015 seconds

Práctica 43

Beleche Alondra

ve\Escritorio\1er Semestre\seminario de progra\p 43.cpp - [Executing] - Dev-C++ 5.11

The screenshot shows the Dev-C++ IDE with a C++ program open and running. The program calculates the product of the diagonal elements of a 4x4 matrix. The output window displays the matrix elements, the matrix itself, and the final product of the diagonal, which is 16.

```
#include <stdio.h>

/* Práctica 43: Producto de La diagonal
P: Beleche Alondra */

int main(int argc, char *argv[]) {
    int matriz[4][4], pro = 1, i, j;

    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            printf("Elemento [%d][%d]: ", i, j);
            scanf("%d", &matriz[i][j]);
        }
    }

    printf("Matriz:\n");
    for (i = 0; i < 4; i++) {
        for (j = 0; j < 4; j++) {
            printf("%d\t", matriz[i][j]);
        }
        printf("\n");
    }

    for (i = 0; i < 4; i++) {
        pro *= matriz[i][i];
    }

    printf("Producto de la diagonal: %d\n", pro);

    return 0;
}
```

Elemento [0][0]: 1
Elemento [0][1]: 2
Elemento [0][2]: 3
Elemento [0][3]: 4
Elemento [1][0]: 5
Elemento [1][1]: 4
Elemento [1][2]: 3
Elemento [1][3]: 2
Elemento [2][0]: 3
Elemento [2][1]: 2
Elemento [2][2]: 2
Elemento [2][3]: 2
Elemento [3][0]: 2
Elemento [3][1]: 2
Elemento [3][2]: 2
Elemento [3][3]: 2
Matriz:
1 2 3 4
5 4 3 2
3 2 2 2
2 2 2 2
Producto de la diagonal: 16

Process exited after 11.73 seconds with return value 0
Presione una tecla para continuar . . .

Registro de Compilación Depuración Resultados Cerrar

Compilation Time: 0.41s

Sel: 0 Lines: 32 Length: 684 Insertar Done parsing in 0.015 seconds

Práctica 44

Beleche Alondra

ve\Escritorio\1er Semetre\seminario de progra\p 44.cpp - [Executing] - Dev-C++ 5.11

The image shows the Dev-C++ IDE with a C++ program for matrix multiplication. The program defines three 3x3 matrices: m1, m2, and m3. It prompts the user to enter elements for m1 and m2, then calculates the product m3 = m1 * m2. The output displays the elements of m1, m2, and the resulting matrix m3.

```
1 #include <stdio.h>
2
3 /* Práctica 44: Multiplicación de matrices
4  P: Beleche Alondra */
5
6 int main(int argc, char *argv[]) {
7     int m1[3][3], m2[3][3], m3[3][3]={0}, i, j, k;
8
9     for (i=0; i<3; i++) {
10         for (j=0; j<3; j++) {
11             printf("Elemento [%d][%d] matriz 1: ", i, j);
12             scanf("%d", &m1[i][j]);
13         }
14         printf("\n");
15     }
16     printf("\n");
17
18     for (i=0; i<3; i++) {
19         for (j=0; j<3; j++) {
20             printf("Elemento [%d][%d] matriz 2: ", i, j);
21             scanf("%d", &m2[i][j]);
22         }
23     }
24
25     for (i=0; i<3; i++) {
26         for (j=0; j<3; j++) {
27             for (k=0; k<3; k++) {
28                 m3[i][j] += m1[i][k] * m2[k][j];
29             }
30         }
31     }
32
33     printf("\nMatriz resultante:\n");
34
35     for (i=0; i<3; i++) {
36         for (j=0; j<3; j++) {
37             printf("%d\t", m3[i][j]);
38         }
39         printf("\n");
40     }
41
42     return 0;
43 }
44
45
```

Execution Output:

```
Elemento [0][2] matriz 1: 7
Elemento [1][0] matriz 1: 5
Elemento [1][1] matriz 1: 4
Elemento [1][2] matriz 1: 3

Elemento [2][0] matriz 1: 2
Elemento [2][1] matriz 1: 3
Elemento [2][2] matriz 1: 5

Elemento [0][0] matriz 2: 6
Elemento [0][1] matriz 2: 7
Elemento [0][2] matriz 2: 8
Elemento [1][0] matriz 2: 9
Elemento [1][1] matriz 2: 0
Elemento [1][2] matriz 2: 0
Elemento [2][0] matriz 2: 1
Elemento [2][1] matriz 2: 1
Elemento [2][2] matriz 2: 2

Matriz resultante:
127    63    78
69     38    46
44     19    26

-----
Process exited after 17.35 seconds with return value 0
Presione una tecla para continuar . . .
```

Bottom status bar: Sel: 0 Lines: 45 Length: 995 Insertar Done parsing in 0.031 seconds

// Práctica 39, llenar arreglos

• Librerías

Principal

Inicio

```
int matriz [4][4] = {22,45,129,36}, {98,7,4,514},  
{123,46,47,41}, {12,78,38} };
```

```
int i, j;
```

```
for (i = 0; i < 4; i++)
```

```
{
```

```
    for (j = 0; j < 4; j++)
```

```
{
```

```
        printf("Dame el elemento %d,%d del arreglo \n", i, j);
```

```
        scanf("%d", &matriz[i][j]);
```

```
    }
```

```
}
```

```
for (i = 0; i < 4; i++)
```

```
{
```

```
    for (j = 0; j < 4; j++)
```

```
{
```

```
        printf("El elemento %d,%d es %d \n", i, j, matriz[i][j]);
```

```
    }
```

```
}
```

```
return 0;
```

Fin

Beleche Mendoza Alondra

D

M

A

30

04

22

Scribe

// Práctica 40, Llena arreglos.

• Librería
Principal
Inicio

```
int A1[3][3], B1[3][3], C1[3][3], i, j;  
for (i = 0; i < 3; i++) {  
    for (j = 0; j < 3; j++) {  
        imprimir ("Dame el elemento %d, %d del arreglo A1", i, j);  
        leer ("%d", &A1[i][j]);  
    }  
}  
for (i = 0; i < 3; i++) {  
    for (j = 0; j < 3; j++) {  
        C1[i][j] = A1[i][j] + B1[i][j];  
        imprimir ("%d + %d = %d", A1[i][j], B1[i][j], C1[i][j]);  
    }  
}  
return 0;  
Fin
```

#include <stdio.h> → Librerías

#include <stdlib.h>

// Arreglos unidimensionales y bidimensionales. Práctica 41

int main () → principal

{ → Inicializar el programa

int x[3][3] = {7, 15, 3, 4, 9, 11, 6, 2, 5}, y[3] = {0, 0, 0},

i, j, m = 1 → Declaración de variables

* Comienza llenando por columnas

for (i = 0; i < 3; i++) → Están anidados los for

{ m = 1; → Inicializa en 1 y continua imprimiendo por

for (j = 0; j < 3; j++) columnas

{ m = m * x[i][j]; → multiplica m * x en la posición 00
y[i] = m; } } pero m no regresa a 1,

se va acumulando

for (i = 0; i < 3; i++) → Inicializa el vector y

{ printf ("%d", y[i]); imprimiendo las filas

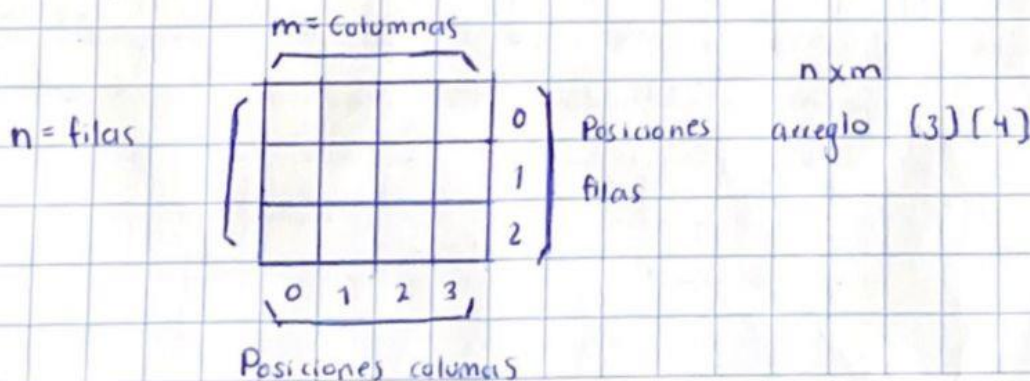
y[i] = m + j; } → Hace la suma de m + j

for (i = 0; i < 3; i++) → Como no tiene llaves solo es

printf ("%d %d", i + 1, y[i]); una instrucción

* Se repite hasta que sea falso

return 0; } → Fin del programa



// Practica 42, Suma de columnas

• Librerías

Principal

Inicio

```
int matriz [3][3], col [3] = {0, 0, 0}, i, j;  
desde (i = 0; i < 3; i++) {  
    desde (j = 0; j < 3; j++) {  
        imprimir ("Elemento [%d] [%d]:", i, j);  
        leer ("%d", &matriz [i][j]);  
        col [j] += matriz [i][j];  
    }  
}  
  
imprimir ("Matriz ingresada:");  
desde (i = 0; i < 3; i++) {  
    desde (j = 0; j < 3; j++) {  
        imprimir ("%d", matriz [i][j]);  
    }  
}  
  
imprimir ("Suma de columnas:");  
desde (j = 0; j < 3; j++) {  
    imprimir ("%d", col [j]);  
}  
  
imprimir ("\n");
```

regresa 0;

Fin

// Practica 43, Producto de la diagonal

- Librerias

Principal

Inicio

```
int matriz [4][4], pro = 1, i, j;
desde (i = 0; i < 4; i++) {
    desde (j = 0; j < 4; j++) {
        imprimir ("Elemento [%d][%d]:", i, j);
        leer ("%d", &matriz [i][j]);
    }
}
```

```
imprimir ("matriz:");
```

```
desde (i = 0; i < 4; i++) {
    desde (j = 0; j < 4; j++) {
```

```
    desde (i = 0; i < 4; i++) {
```

```
        pro *= matriz [i][j];
```

```
    }
```

```
    imprimir ("Producto de la diagonal: %d", pro);
```

regresa 0;

Fin

// Práctica 44. Multiplicación de matrices

- Librerías
- Principal
Inicio

```
int m1[3][3], m2[3][3], mr[3][3] = {0}, i, j, k;  
desde (i=0; i<3; i++) {  
    desde (j=0; j<3; j++) {  
        imprimir ("Elemento [%d][%d] matriz 1:", i, j);  
        leer ("%d", &m1[i][j]);  
    }  
    imprimir (" ----- ");  
    imprimir ("\n");  
    desde (i=0; i<3; i++) {  
        desde (j=0; j<3; j++) {  
            imprimir ("Elemento [%d][%d] matriz 2:", i, j);  
            leer ("%d", &m2[i][j]);  
        }  
    }  
    desde (i=0; i<3; i++) {  
        desde (j=0; j<3; j++) {  
            desde (k=0; k<3; k++) {  
                mr[i][j] += m1[i][k] + m2[k][j];  
            }  
        }  
        imprimir ("Matriz resultante:");  
        desde (i=0; i<3; i++) {  
            desde (j=0; j<3; j++) {  
                imprimir ("%d", mr[i][j]);  
            }  
        }  
    }  
    regresar 0;  
Fin
```