



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Ing. Karina García Morales

*Asignatura:* Fundamentos de la Programación

*Grupo:* 20

*No. de práctica(s):* 10

*Integrante(s):* Martínez Ordoñez Diego Tonatiah

*No. de lista o brigada:* 30

*Semestre:* 2023-1

*Fecha de entrega:* 6 / diciembre / 2022

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# Práctica 10: Arreglos multidimensionales.

## - Objetivo:

El alumno utilizará arreglos de dos dimensiones en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, en estructuras que utilicen dos índices.

## - Desarrollo:

Un arreglo de dos dimensiones es un conjunto de datos contiguos del mismo tipo con un tamaño fijo definido al momento de crearse. Para acceder a un elemento en este tipo de arreglos se requiere el uso de dos índices. Los arreglos se utilizan para hacer más eficiente el código de un programa, ya que la manipulación de datos del mismo tipo que son agrupados en un arreglo por tener un significado común se realiza de una forma más clara y eficaz.

## - Arreglos multidimensionales:

Lenguaje C permite crear arreglos de varias dimensiones con la siguiente sintaxis: `tipoDato nombre[ tamaño ][ tamaño ]...[tamaño];`

Donde nombre se refiere al identificador del arreglo, tamaño es un número entero y define el número máximo de elementos que puede contener el arreglo por dimension.

De manera práctica se puede considerar que la primera dimensión corresponde a los renglones, la segunda a las columnas, la tercera al plano, y así sucesivamente. Sin embargo, en la memoria cada elemento del arreglo se guarda de forma contigua, por lo tanto, se puede recorrer un arreglo multidimensional con apuntadores.

## - Códigos en clase:

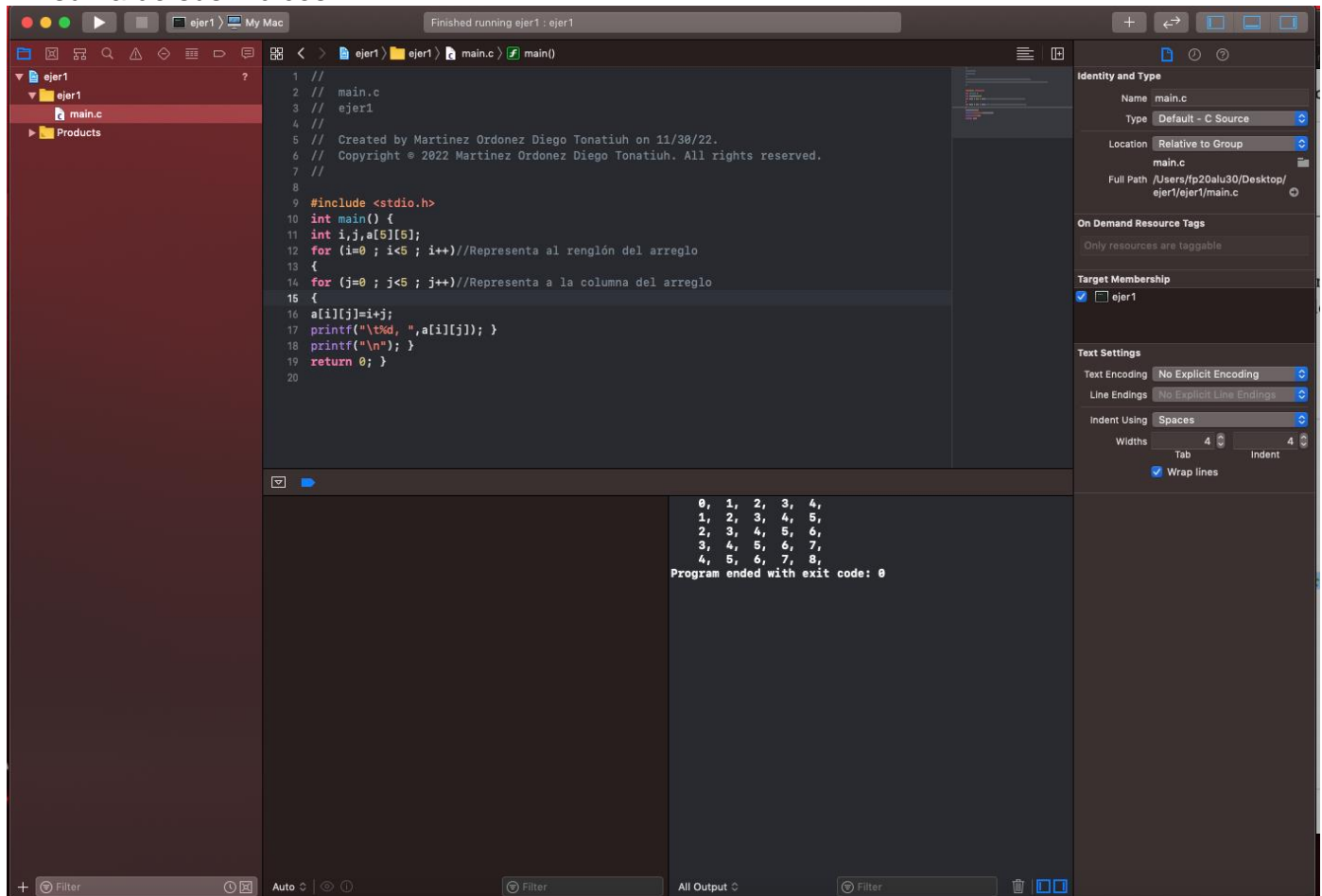
1. Un programa que genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a cada uno de sus elementos por la posición que indica el renglón y la columna a través de dos ciclos for, uno anidado dentro de otro.

```
1 //  
2 // main.c  
3 // ejer1  
4 //  
5 // Created by Martinez Ordonez Diego Tonatiuh on 11/30/22.  
6 // Copyright © 2022 Martinez Ordonez Diego Tonatiuh. All rights reserved.  
7 //  
8  
9 #include <stdio.h>  
10 int main() {  
11     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};  
12     int i, j;  
13     printf("Imprimir Matriz\n");  
14     for (i=0 ; i<3 ; i++) //Representa al renglón del arreglo  
15     {  
16         for (j=0 ; j<3 ; j++)//Representa a la columna del arreglo  
17         {  
18             printf("%d, ",matriz[i][j]); }  
19         printf("\n");  
20     }  
21     return 0;  
22 }  
23
```

Program ended with exit code: 0

1, 2, 3,  
4, 5, 6,  
7, 8, 9,  
Program ended with exit code: 0

2. El siguiente programa genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos por la posición que indica el renglón y la columna a través de dos ciclos for, uno anidado dentro de otro, el contenido de cada elemento de este arreglo es la suma de sus índices.



```
1 //
2 // main.c
3 // ejer1
4 //
5 // Created by Martinez Ordonez Diego Tonatiuh on 11/30/22.
6 // Copyright © 2022 Martinez Ordonez Diego Tonatiuh. All rights reserved.
7 //
8
9 #include <stdio.h>
10 int main() {
11     int i,j,a[5][5];
12     for (i=0 ; i<5 ; i++)//Representa al renglón del arreglo
13     {
14         for (j=0 ; j<5 ; j++)//Representa a la columna del arreglo
15         {
16             a[i][j]=i+j;
17             printf("\t%d", a[i][j]);
18             printf("\n");
19         }
20     }
21     return 0; }
```

0, 1, 2, 3, 4,  
1, 2, 3, 4, 5,  
2, 3, 4, 5, 6,  
3, 4, 5, 6, 7,  
4, 5, 6, 7, 8,  
Program ended with exit code: 0

### - Arreglos multidimensionales usando while.

3. El código que se observa a continuación genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos por la posición que indica el renglón y la columna a través de dos ciclos while, uno anidado dentro de otro.

```
1 // main.c
2 // ejer1
3 //
4 //
5 // Created by Martinez Ordenez Diego Tonatiuh on 11/30/22.
6 // Copyright © 2022 Martinez Ordenez Diego Tonatiuh. All rights reserved.
7 //
8
9 #include <stdio.h>
10 int main() {
11     int matriz[4][4] = {{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}}; int i, j;
12     printf("Imprimir Matriz\n");
13     i=0;
14     while(i<4) //Representa al renglón del arreglo
15     {
16         j=0;
17         while (j<4) //Representa a la columna del arreglo
18         {
19             printf("%d, ",matriz[i][j]);
20             j++; }
21         printf("\n");
22         i++; }
23     return 0; }
24
```

Imprimir Matriz  
1, 2, 3, 4,  
5, 6, 7, 8,  
9, 10, 11, 12,  
13, 14, 15, 16,  
Program ended with exit code: 0

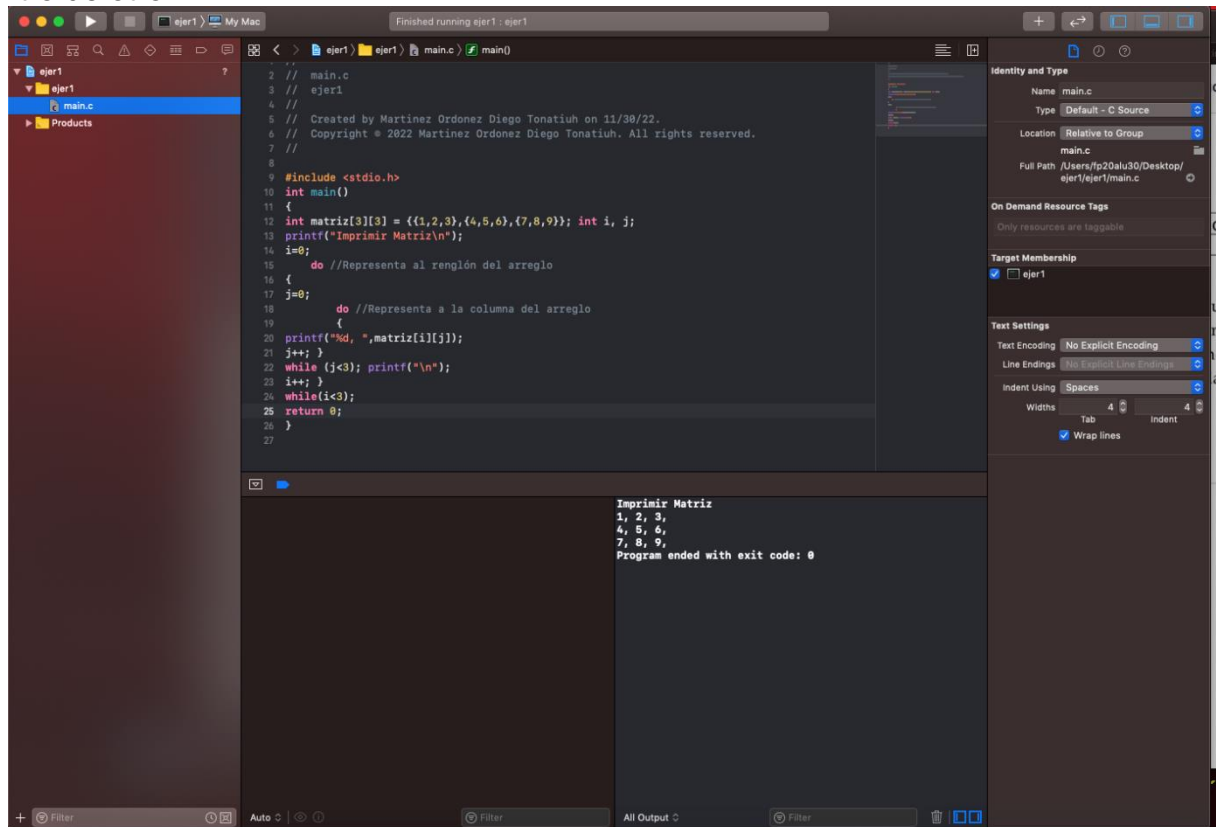
4. Enseguida se muestra el código de un programa que permite generar un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos por la posición que indica el renglón y la columna a través de dos ciclos while, uno anidado dentro de otro, el contenido de cada elemento de este arreglo es la suma de sus índices.

```
1 // main.c
2 // ejer1
3 //
4 //
5 // Created by Martinez Ordenez Diego Tonatiuh on 11/30/22.
6 // Copyright © 2022 Martinez Ordenez Diego Tonatiuh. All rights reserved.
7 //
8
9 #include <stdio.h>
10 int main()
11 {
12     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}}; int i, j;
13     printf("Imprimir Matriz\n");
14     i=0;
15     do //Representa al renglón del arreglo
16     {
17         j=0;
18         do //Representa a la columna del arreglo
19         {
20             printf("%d, ",matriz[i][j]);
21             j++; }
22         while (j<3); printf("\n");
23         i++; }
24     while(i<3);
25     return 0;
26 }
27
```

Imprimir Matriz  
1, 2, 3,  
4, 5, 6,  
7, 8, 9,  
Program ended with exit code: 0

## - Arreglos multidimensionales usando do-while.

5. A continuación, se presenta un código que permite la generación de un arreglo de dos dimensiones (arreglo multidimensional) y se puede acceder a cada uno de sus elementos por la posición que indica el renglón y la columna a través de dos ciclos do-while, uno anidado dentro de otro.



```
1 // main.c
2 // ejer1
3 //
4 //
5 // Created by Martinez Ordóñez Diego Tonatiuh on 11/30/22.
6 // Copyright © 2022 Martinez Ordóñez Diego Tonatiuh. All rights reserved.
7 //
8
9 #include <stdio.h>
10 int main()
11 {
12     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}}; int i, j;
13     printf("Imprimir Matriz\n");
14     i=0;
15     do //Representa al renglón del arreglo
16     {
17         j=0;
18         do //Representa a la columna del arreglo
19         {
20             printf("%d, ",matriz[i][j]);
21             j++;
22         } while (j<3); printf("\n");
23         i++;
24     } while(i<3);
25     return 0;
26 }
27
```

Imprimir Matriz  
1, 2, 3,  
4, 5, 6,  
7, 8, 9,  
Program ended with exit code: 0

6. Un arreglo de dos dimensiones (arreglo multidimensional) y se accede a todos sus elementos por la posición que indica el renglón y la columna a través de dos ciclos do-while

The screenshot shows the Xcode IDE with a C program named `main.c` in a project called `ejer1`. The code defines a 5x5 array `a` and uses nested loops to print its contents. The output window shows the array values and the program's exit code.

```
1 //
2 // main.c
3 // ejer1
4 //
5 // Created by Martinez Ordenez Diego Tonatiuh on 11/30/22.
6 // Copyright © 2022 Martinez Ordenez Diego Tonatiuh. All rights reserved.
7 //
8
9 #include <stdio.h>
10 int main() {
11     int i,j,a[5][5];
12     i=0;
13     do //Representa al renglón del arreglo
14     {
15         j=0;
16         do //Representa a la columna del arreglo
17         {
18             a[i][j]=i+j;
19             printf("\td", "a[i][j]");
20             j++;
21         }
22         while (j<5);
23         printf("\n");
24         i++;
25     }
26     while (i<5); return 0;
27 }
28
29
```

Output:

```
0, 1, 2, 3, 4,
1, 2, 3, 4, 5,
2, 3, 4, 5, 6,
3, 4, 5, 6, 7,
4, 5, 6, 7, 8,
Program ended with exit code: 0
```

7. Un programa que genera un arreglo multidimensional de máximo 10 renglones y 10 columnas, para poder almacenar datos en cada elemento y posteriormente mostrar el contenido de esos elementos se hace uso de ciclos for anidados respectivamente.

The screenshot shows the Xcode IDE with a C program named `main.c` in a project called `ejer1`. The code defines a 3x3 array `lista` and uses nested loops to print its contents. The output window shows the array values and the program's exit code.

```
1 //
2 // main.c
3 // ejer1
4 //
5 // Created by Martinez Ordenez Diego Tonatiuh on 11/30/22.
6 // Copyright © 2022 Martinez Ordenez Diego Tonatiuh. All rights reserved.
7 //
8
9 #include <stdio.h>
10 int main ()
11 {
12     int lista[3][3]; // Se declara el arreglo multidimensional
13     int i,j;
14     int renglon,columna;
15     printf("\nDa el número de renglones y columnas separados con coma\n");
16     scanf("%d,%d",&renglon,&columna);
17     if(((renglon>=1) && (renglon<=3)&&(columna>=1) && (columna<=3)))
18     {
19         // Acceso a cada elemento del arreglo multidimensional usando for
20         for (i= 0 ; i <= renglon-1 ; i++)
21         {
22             for(j= 0 ; j <= columna-1 ; j++)
23             {
24                 printf("\nNúmero para el elemento %d,%d del arreglo", i,j );
25                 scanf("%d",&lista[i][j]);
26             }
27         }
28         printf("\nLos valores dados son: \n");
29         // Acceso a cada elemento del arreglo multidimensional usando for
30         for (i= 0 ; i <= renglon-1 ; i++)
31         {
32             for(j= 0 ; j <= columna-1 ; j++)
33             {
34                 printf("%d", lista[i][j]);
35             }
36         }
37     }
38 }
```

Output:

```
Da el número de renglones y columnas separados con coma
2
Número para el elemento 0,0 del arreglo1
Número para el elemento 1,0 del arreglo5
Los valores dados son:
1
5
Program ended with exit code: 0
```

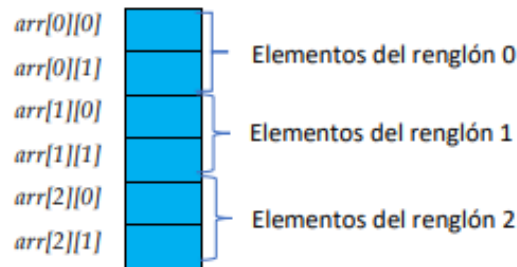
## - Apuntadores y su relación con arreglos de dos dimensiones.

La sintaxis para declarar un apuntador y para asignarle la dirección de memoria de otra variable es, respectivamente:

```
TipoDeDato *apuntador, variable;  
apuntador = &variable;
```

La declaración de una variable apuntador inicia con el carácter \*. Cuando a una variable le antecede un ampersand (&), lo que se hace es referirse a la dirección de memoria donde se ubica el valor de dicha variable (es lo que pasa cuando se lee un dato con scanf).

Los apuntadores solo pueden apuntar a direcciones de memoria del mismo tipo de dato con el que fueron declarados; para acceder al contenido de dicha dirección, a la variable apuntador se le antepone \*.



## - Arreglos multidimensionales con apuntadores.

8. El programa siguiente genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos a través de un apuntador utilizando un ciclo for.

```
1 #include<stdio.h>  
2 int main()  
3 {  
4     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};  
5     int i, cont=0, *ap;  
6     ap = "matriz; //Esta sentencia es análoga a: ap = &matriz[0][0];  
7     printf("Imprimir Matriz\n");  
8     for (i=0 ; i<9 ; i++)  
9     {  
10        if (cont == 3) //Se imprimió un renglón y se hace un salto de línea  
11        {  
12            printf("\n");  
13            cont = 0; //Inicia conteo de elementos del siguiente renglón  
14        }  
15        printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz  
16        cont++;  
17    }  
18    printf("\n");  
19    return 0;  
20 }
```

Imprimir Matriz

1	2	3
4	5	6
7	8	9

Process exited after 0.5493 seconds with return value 0  
Presione una tecla para continuar . . .

9. El código del siguiente programa genera un arreglo de dos dimensiones (arreglo multidimensional) y accede a sus elementos a través de un apuntador utilizando un ciclo while.



The screenshot shows the Dev-C++ IDE with a C++ program named `practica9.cpp` being executed. The program defines a 3x3 matrix and uses a `while` loop to iterate through its elements. A separate window displays the output, showing the matrix and a confirmation message.

```
1 #include<stdio.h>
2 int main()
3 {
4     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
5     int i, cont=0, *ap;
6     ap = *matriz; //Esta sentencia es andloga a: ap = &matriz[0][0];
7     printf("Imprimir Matriz\n");
8     i=0;
9     while (i<9)
10    {
11        if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
12        {
13            printf("\n");
14            cont = 0; //Inicia conteo de elementos del siguiente renglón
15        }
16        printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
17        cont++;
18        i++;
19    }
20    printf("\n");
21    return 0;
22 }
```

Output window content:

```
Imprimir Matriz
1   2   3
4   5   6
7   8   9

-----
Process exited after 0.3096 seconds with return value 0
Presione una tecla para continuar . . .
```

10. La generación de un arreglo de dos dimensiones (arreglo multidimensional) y el acceso a sus elementos a través de un apuntador utilizando un ciclo do-while se puede observar en el programa siguiente:

This screenshot shows the same C++ program as above, but using a `do-while` loop for iteration. The output window shows the matrix and execution details. Below the IDE, the 'Compilation results...' window is visible, showing no errors or warnings.

```
1 #include<stdio.h>
2 int main()
3 {
4     int matriz[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
5     int i, cont=0, *ap;
6     ap = *matriz; //Esta sentencia es andloga a: ap = &matriz[0][0];
7     printf("Imprimir Matriz\n");
8     i=0;
9     do
10    {
11        if (cont == 3) //Se imprimió un renglón y se hace un salto de línea
12        {
13            printf("\n");
14            cont = 0; //Inicia conteo de elementos del siguiente renglón
15        }
16        printf("%d\t",*(ap+i)); //Se imprime el siguiente elemento de la matriz
17        cont++;
18        i++;
19    } while (i<9);
20    printf("\n");
21    return 0;
22 }
```

Output window content:

```
Imprimir Matriz
1   2   3
4   5   6
7   8   9

-----
Process exited after 0.3464 seconds with return value 0
Presione una tecla para continuar . . .
```

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\diego\OneDrive\Escritorio\Universidad\Fundamentos de la programacion\Programas\practica9.exe
- Output Size: 128.7724609375 KiB
- Compilation Time: 66.13s



- Tabla  
 Valor inicial -----  
 Condición -----  
 Incremento -----

For	while	Do-while
<pre>#include&lt;stdio.h&gt; int main() {   int i,j,a[5][5];   for (i=0 ; i&lt;5 ; i++)//Representa al renglón   del arreglo   {     for (j=0 ; j&lt;5 ; i++)//Representa a la     columna del arreglo     {       a[i][j]=i+j;       printf("\t%d, ",a[i][j]);     }     printf("\n");   }   return 0; }</pre>	<pre>#include&lt;stdio.h&gt; int main() {   int i,j,a[5][5];   i=0;   while (i&lt;5) //Representa al   renglón del arreglo   {     j=0;     while (j&lt;5) //Representa a la     columna del arreglo+     {       a[i][j]=i+j;       printf("\t%d, ",a[i][j]);       j++;     }     printf("\n");     i++;   }   return 0; }</pre>	<pre>#include&lt;stdio.h&gt; int main() {   int i,j,a[5][5];   i=0;   do //Representa al renglón   del arreglo   {     j=0;     do //Representa a la     columna del arreglo     {       a[i][j]=i+j;       printf("\t%d, ",a[i][j]);       j++;     }     while (i&lt;5);     printf("\n");     i++;   }   while (i&lt;5);   return 0; }</pre>
<pre>#include&lt;stdio.h&gt; int main() {   int matriz[3][3] =   {{1,2,3},{4,5,6},{7,8,9}};   int i, cont=0, *ap;   ap = *matriz; //Esta   sentencia es análoga a: ap =   &amp;matriz[0][0];   printf("Imprimir Matriz\n");   for (i=0 ; i&lt;9 ; i++)   {     if (cont == 3) //Se imprimió     un renglón y se hace un salto     de línea     {       printf("\n");       cont = 0; //Inicia conteo de       elementos del siguiente       renglón     }     printf("%d\t",*(ap+i)); //Se     imprime el siguiente     elemento de la matriz     cont++;   } }</pre>	<pre>#include&lt;stdio.h&gt; int main() {   int matriz[3][3] =   {{1,2,3},{4,5,6},{7,8,9}};   int i, cont=0, *ap;   ap = *matriz; //Esta sentencia   es análoga a: ap =   &amp;matriz[0][0];   printf("Imprimir Matriz\n");   i=0;   while (i&lt;9)   {     if (cont == 3) //Se imprimió     un renglón y se hace un salto     de línea     {       printf("\n");       cont = 0; //Inicia conteo de       elementos del siguiente       renglón     }     printf("%d\t",*(ap+i)); //Se     imprime el siguiente     elemento de la matriz     i++;   } }</pre>	<pre>#include&lt;stdio.h&gt; int main() {   int matriz[3][3] =   {{1,2,3},{4,5,6},{7,8,9}};   int i, cont=0, *ap;   ap = *matriz; //Esta   sentencia es análoga a: ap =   &amp;matriz[0][0];   printf("Imprimir Matriz\n");   i=0;   do   {     if (cont == 3) //Se imprimió     un renglón y se hace un     salto de línea     {       printf("\n");       cont = 0; //Inicia conteo de       elementos del siguiente       renglón     }     printf("%d\t",*(ap+i)); //Se     imprime el siguiente     elemento de la matriz     cont++;   }   while (i&lt;9); }</pre>

<pre> } printf("\n"); return 0; } </pre>	<pre> cont++; i++; } printf("\n"); return 0; } </pre>	<pre> cont++; i++; } while (i&lt;9); printf("\n"); return 0; } </pre>
--	---	---

### - Conclusiones:

Esta práctica nos ayudó mucho en el ámbito general de los conocimientos, gracias a que nos brindó nuevos conceptos de como podemos aplicar algunos métodos para poder ejecutarlos de mejor manera.

De igual forma, el cómo podemos usar diferentes tipos de estructuras para diferentes códigos y nos darán el mismo resultado de manera en que podemos tener más de una solución para este tipo de trabajos, solo es cuestión de ponerlas en práctica para poder mejorar en este apartado. Conocemos más herramientas de apoyo, como son los arreglos y como podemos diferenciarlos, por otro lado, el cómo nos van apoyar en la cuestión académica y de cierta manera, en un ambiente laboral donde tengamos que poner a prueba lo aprendido.

### - Bibliografía:

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Educación 1991.

# TAREA DE ARREGLO MULTIDIMENSIONAL

Alumno: Martínez Ordoñez Diego Tonatiuh

## 1. Programa con nombre y número de cuenta

The screenshot displays the PSeInt IDE interface. The main editor window contains the following pseudocode:

```
1 Algoritmo NombreCuenta
2   definir nom, cuenta Como Caracter
3   escribir "Su nombre completo es..."
4   leer nom
5   escribir "Tú número de cuenta es..."
6   leer cuenta
7   escribir ("Tú nombre es:" + nom)
8   escribir ("Tú número de cuenta es:" + cuenta)
9   FinAlgoritmo
10
```

An execution window titled "PSeInt - Ejecutando proceso NOMBRECuenta" shows the program's output:

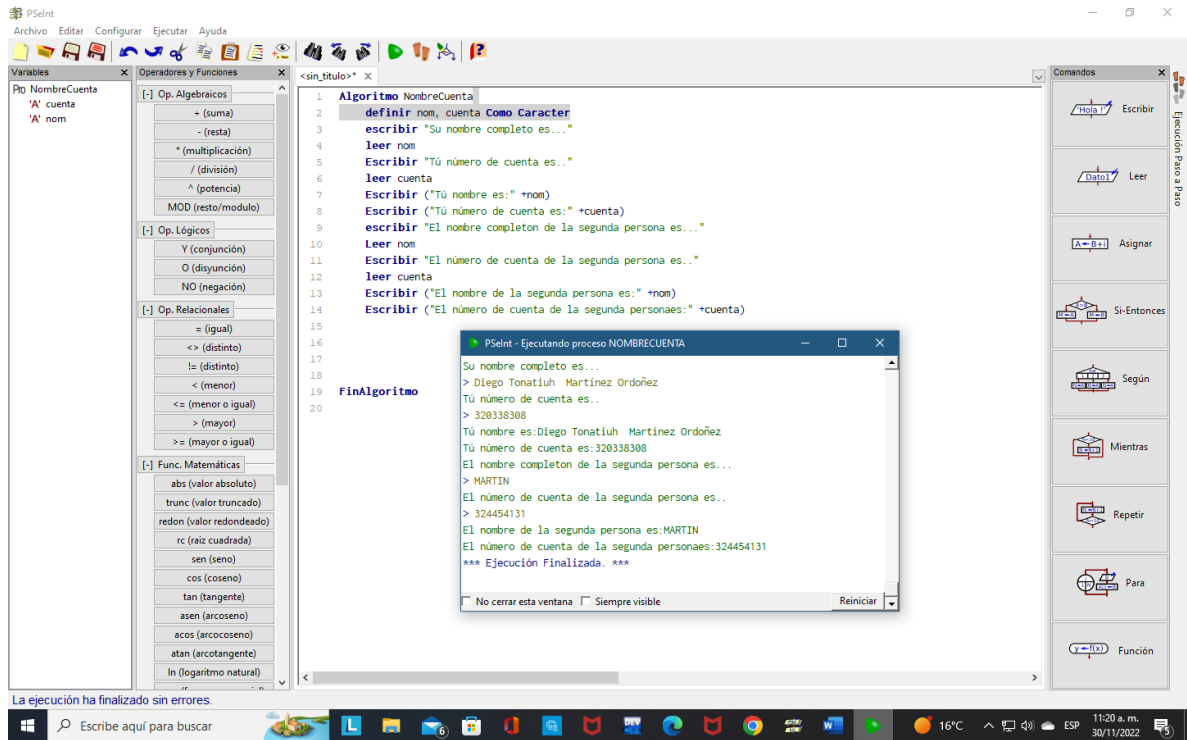
```
*** Ejecución Iniciada. ***
Su nombre completo es...
> Diego Tonatiuh Martínez Ordoñez
Tú número de cuenta es..
> 320338308
Tú nombre es:Diego Tonatiuh Martínez Ordoñez
Tú número de cuenta es:320338308
*** Ejecución Finalizada. ***
```

On the right side, the "Comandos" (Commands) panel lists various control flow symbols: Escribir, Leer, Asignar, Si-Entonces, Según, Mientras, Repetir, Para, and Función.

At the bottom, the "Errores de Sintaxis" (Syntax Errors) panel displays the message: "La sintaxis es correcta. Puede presionar F9 para ejecutar el algoritmo."

The Windows taskbar at the bottom shows the system clock as 11:02 a.m. on 30/11/2022, with a temperature of 16°C.

## 2. Con apuntadores



## 3. Programa

