



UNRN

Universidad Nacional
de Río Negro



| unrionegro

El lenguaje de programación C [IV]

UNRN

Universidad Nacional
de **Río Negro**

El equipo de Programación I

Arreglos Unidimensionales

tipo nombre[tamano];

Declaración

```
int arreglo[5];  
for (int i = 0; i < 5; i++) {  
    arreglo[i] = i;  
}
```

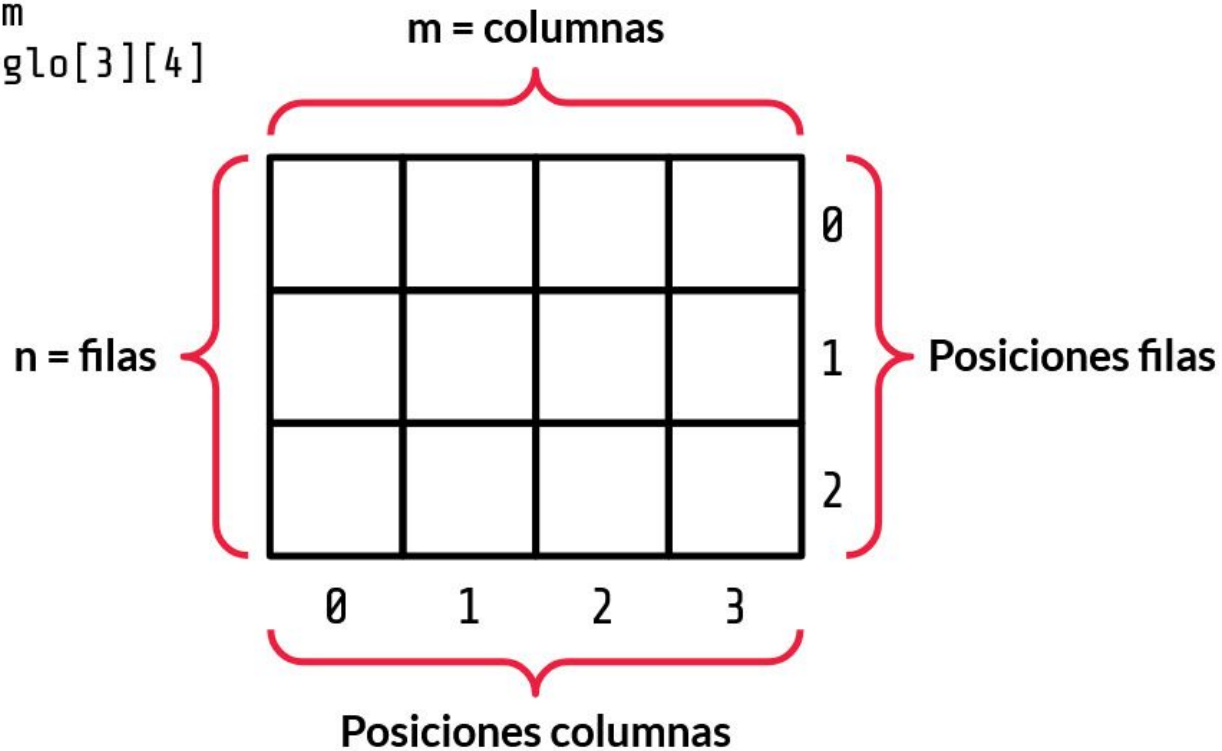
0	1	2	3	4
---	---	---	---	---



¿Preguntas?

Arreglos Bidimensionales (matrices)

$n \times m$
`arreglo[3][4]`




```
tipo nombre[tamano_fila][tamano_columna];
```

Carga de una matriz

```
int numeros[3][3];
```

```
for (i=0;i<3;i++)  
{  
    for (j=0;j<3;j++)  
    {  
        numeros[i][j] = sum++;  
    }  
}
```

0	1	2
3	4	5
6	7	8

Impresión de una matriz

```
for (i=0;i<3;i++)  
{  
    printf("\n");  
    for (j=0;j<3;j++)  
        printf("[%d]", numeros[i][j]);  
}
```

```
printf  
[0][1][2]  
[3][4][5]  
[6][7][8]
```

0	1	2
3	4	5
6	7	8



¿Preguntas?

EJ: Matriz traspuesta

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \rightarrow A^T = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 11 & 12 & 13 \end{pmatrix} \rightarrow A^T = \begin{pmatrix} 1 & 11 \\ 2 & 12 \\ 3 & 13 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \rightarrow A^T = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow A^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

EJ: Matriz traspuesta

MATRIZ

0	1	2
3	4	5
6	7	8

TRASPUESTA

0	3	6
1	4	7
2	5	8

EJ: Matriz traspuesta

```
for (i=0;i<3;i++){  
    printf("\n");  
    for (j=0;j<3;j++){  
        printf("%d",numeros[i][j]);}
```

0	1	2
3	4	5
6	7	8

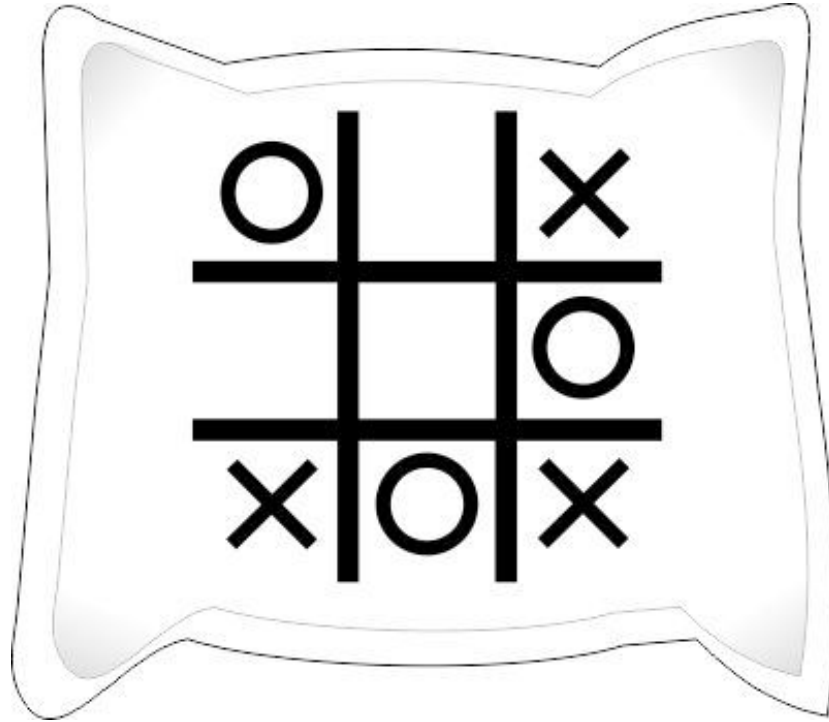
```
for (i=0;i<3;i++){  
    printf("\n");  
    for (j=0;j<3;j++){  
        printf("%d",numeros[j][i]);}
```

0	3	6
1	4	7
2	5	8



¿Preguntas?

Ejercicio: TATETI



CASOS:

Ejercicio: TATETI

[x]	[o]	[o]
[x]	[o]	[]
[x]	[]	[]

[]	[x]	[o]
[]	[x]	[]
[o]	[x]	[o]

[]	[o]	[x]
[o]	[o]	[x]
[x]	[]	[x]

[x]	[x]	[x]
[]	[o]	[]
[x]	[]	[]

[]	[x]	[o]
[x]	[x]	[x]
[o]	[]	[o]

[]	[o]	[x]
[o]	[o]	[]
[x]	[x]	[x]

CASOS: Ejercicio: TATETI

[x]	[o]	[o]
[]	[x]	[]
[x]	[]	[x]

[o]	[]	[x]
[]	[x]	[o]
[x]	[x]	[o]



¿Preguntas?

Estructuras de información (structs)

Declaración

```
struct fraccion {  
    int numerador;  
    int denominador;  
};
```

Inicialización

```
struct fraccion f2 = {1, 4};
```

Inicialización

(más explícita)

```
struct fraccion f3 = { .denominador = 1, .numerador = 3 };
```


Uso y acceso a miembros

```
struct fraccion f1;  
f1.numerador = 10;  
f1.denominador = 10;
```



Otro ejemplo

```
struct config_puerto_serie periferico;  
  
periferico.baudrate = 115200;  
periferico.wordlength = UART_WORDLENGTH_8B;  
periferico.stopbits = UART_STOPBITS_1;  
periferico.parity = UART_PARITY_NONE;
```



¿Preguntas?

Operaciones con structs

Copia de estructuras

```
struct fraccion f1 = {1, 4};  
struct fraccion f2 = f1;
```

¿Comparación directa?

f2 == f1

no funciona

Miembro a miembro sí

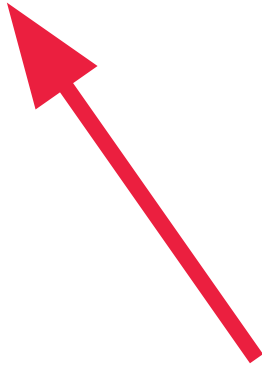
f1.denominador > f2.denominador



¿Preguntas?

Uso en funciones

```
struct fraccion suma_fraccion(struct fraccion frac, int numero);
```



Con esto en mente...

```
struct division {  
    int cociente;  
    int resto;  
};
```

```
struct division division_lenta(int dividendo, int divisor);
```



¿Preguntas?

Sobrenombres (alias)



```
typedef struct fraccion fraccion_t;
```

```
fraccion_t f3;
```

Declaración y alias

```
typedef struct {  
    int cociente;  
    int resto;  
} division_t;
```



¿Preguntas?

fraccion_t multiple[MAX];

Ejercicio: TATETI

¿Lo adaptamos con estructuras?



¿Preguntas?

unrn.edu.ar

UNRN

Universidad Nacional
de **Río Negro**



| **unrionegro**