



ARRAYS

MULTIDIMENSIONAL

Oscar Paniagua @2023

array@multidimensional

```
for (i = 0; i < 2; i++)  
{  
    for (j = 0; j < 3; j++)  
    {  
        printf("%d ", p[i][j]);  
    }  
}
```

SALIDA: 0

	0	1	2
0	0	1	2
1	3	4	5

i = 0	j = 0	p[0][0]	0
-------	-------	---------	---

i = 0	j = 1	p[0][1]	1
-------	-------	---------	---

i = 0	j = 2	p[0][2]	2
-------	-------	---------	---

i = 1	j = 0	p[1][0]	3
-------	-------	---------	---

i = 1	j = 1	p[1][1]	4
-------	-------	---------	---

i = 1	j = 2	p[1][2]	5
-------	-------	---------	---

array@multidimensional

```
for (i = 0; i < 2; i++)
{
    for (j = 0; j < 3; j++)
    {
        printf("%d ", p[i][j]);
    }
}
```

SALIDA: 0 1

	0	1	2
0	0	^{[0][1]} 1	2
1	3	4	5

i = 0	j = 0	p[0][0]	0
-------	-------	---------	---

i = 0	j = 1	p[0][1]	1
-------	-------	---------	---

i = 0	j = 2	p[0][2]	2
-------	-------	---------	---

i = 1	j = 0	p[1][0]	3
-------	-------	---------	---

i = 1	j = 1	p[1][1]	4
-------	-------	---------	---

i = 1	j = 2	p[1][2]	5
-------	-------	---------	---

array@multidimensional

```
for (i = 0; i < 2; i++)  
{  
    for (j = 0; j < 3; j++)  
    {  
        printf("%d ", p[i][j]);  
    }  
}
```

SALIDA: 0 1 2

	0	1	2
0	0	1	^{[0][2]} 2
1	3	4	5

i = 0 j = 0 p[0][0] 0

i = 0 j = 1 p[0][1] 1

i = 0 j = 2 p[0][2] 2

i = 1 j = 0 p[1][0] 3

i = 1 j = 1 p[1][1] 4

i = 1 j = 2 p[1][2] 5

array@multidimensional

```
for (i = 0; i < 2; i++)
{
    for (j = 0; j < 3; j++)
    {
        printf("%d ", p[i][j]);
    }
}
```

SALIDA: 0 1 2 3

	0	1	2
0	0	1	2
1	3	4	5

Note: In the original image, the cell containing '3' at row 1, column 0 is circled in green and labeled [1][0].

i = 0 j = 0 p[0][0] 0

i = 0 j = 1 p[0][1] 1

i = 0 j = 2 p[0][2] 2

i = 1 j = 0 p[1][0] 3

i = 1 j = 1 p[1][1] 4

i = 1 j = 2 p[1][2] 5

array@multidimensional

```
for (i = 0; i < 2; i++)  
{  
    for (j = 0; j < 3; j++)  
    {  
        printf("%d ", p[i][j]);  
    }  
}
```

SALIDA: 0 1 2 3 4

	0	1	2
0	0	1	2
1	3	^{[1][1]} 4	5

i = 0 j = 0 p[0][0] 0

i = 0 j = 1 p[0][1] 1

i = 0 j = 2 p[0][2] 2

i = 1 j = 0 p[1][0] 3

i = 1 j = 1 p[1][1] 4

i = 1 j = 2 p[1][2] 5

array@multidimensional

```
for (i = 0; i < 2; i++)
{
    for (j = 0; j < 3; j++)
    {
        printf("%d ", p[i][j]);
    }
}
```

SALIDA: 0 1 2 3 4 5

	0	1	2
0	0	1	2
1	3	4	^{[1][2]} 5

i = 0 j = 0 p[0][0] 0

i = 0 j = 1 p[0][1] 1

i = 0 j = 2 p[0][2] 2

i = 1 j = 0 p[1][0] 3

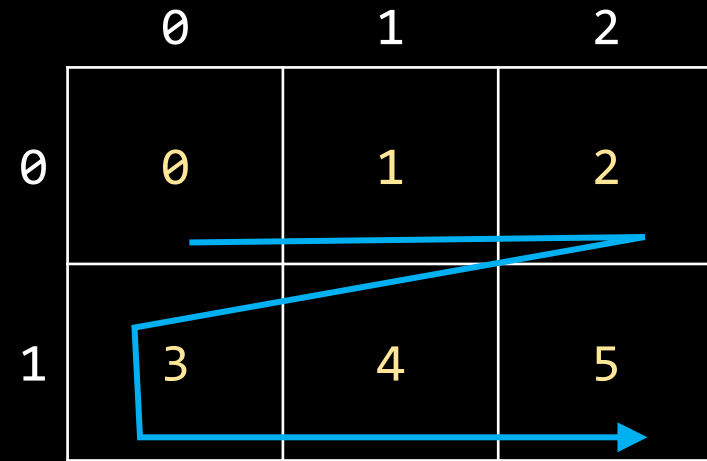
i = 1 j = 1 p[1][1] 4

i = 1 j = 2 p[1][2] 5

array@multidimensional

```
for (i = 0; i < 2; i++)  
{  
    for (j = 0; j < 3; j++)  
    {  
        printf("%d ", p[i][j]);  
    }  
}
```

SALIDA: 0 1 2 3 4 5



i = 0	j = 0	p[0][0]	0
i = 0	j = 1	p[0][1]	1
i = 0	j = 2	p[0][2]	2
i = 1	j = 0	p[1][0]	3
i = 1	j = 1	p[1][1]	4
i = 1	j = 2	p[1][2]	5