

CCP130

Desenvolvimento de Algoritmos

Prof. Danilo H. Perico



Arrays



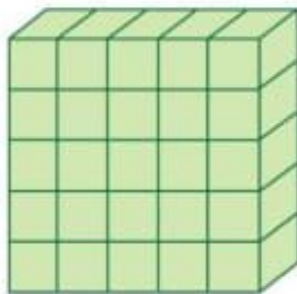
Arrays

- Arranjo (array)
- Conjunto **finito** de elementos **homogêneos**
 - É possível identificar o primeiro, o segundo, ..., o n -ésimo elemento do arranjo
 - Homogeneidade: todos os elementos do arranjo são do mesmo tipo
- Ocupa um grupo de posições de memória adjacentes e identificadas pelo mesmo nome

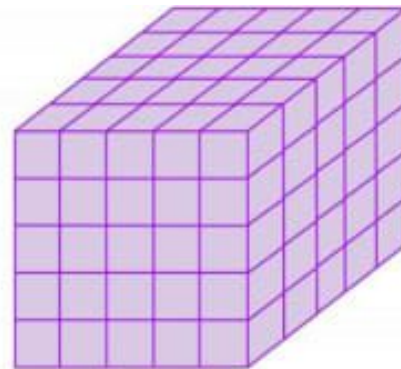
Arrays



vector



matrix



3-D Tensor

Matrices



Matrizes

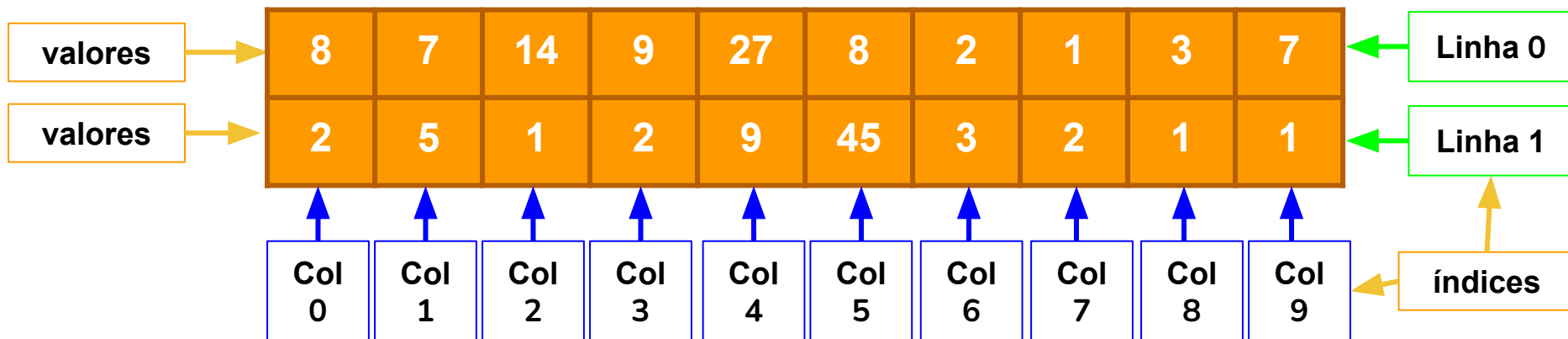
- **Arranjo de DUAS dimensões**
- Exemplo: Matriz de inteiros
 - **2 linhas**
 - **10 colunas**

8	7	14	9	27	8	2	1	3	7
2	5	1	2	9	45	3	2	1	1



Matrizes

- Necessitam de duas **informações posicionais**
 - São **indexadas** por **linha** e **coluna**





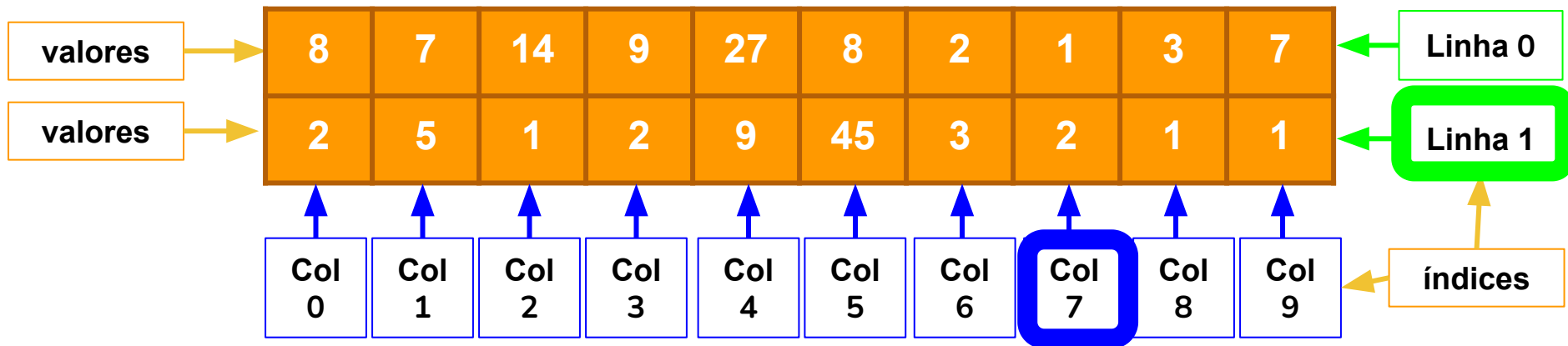
Matrizes

- Necessitam de apenas uma **informação posicional**
 - São **indexados**
- Exemplo: **Qual é o valor no índice - linha 1 e coluna 7?**

valores	→	8	7	14	9	27	8	2	1	3	7	←	Linha 0
valores	→	2	5	1	2	9	45	3	2	1	1	←	Linha 1
		↑	↑	↑	↑	↑	↑	↑	↑	↑	↑		
		Col 0	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	←	índices

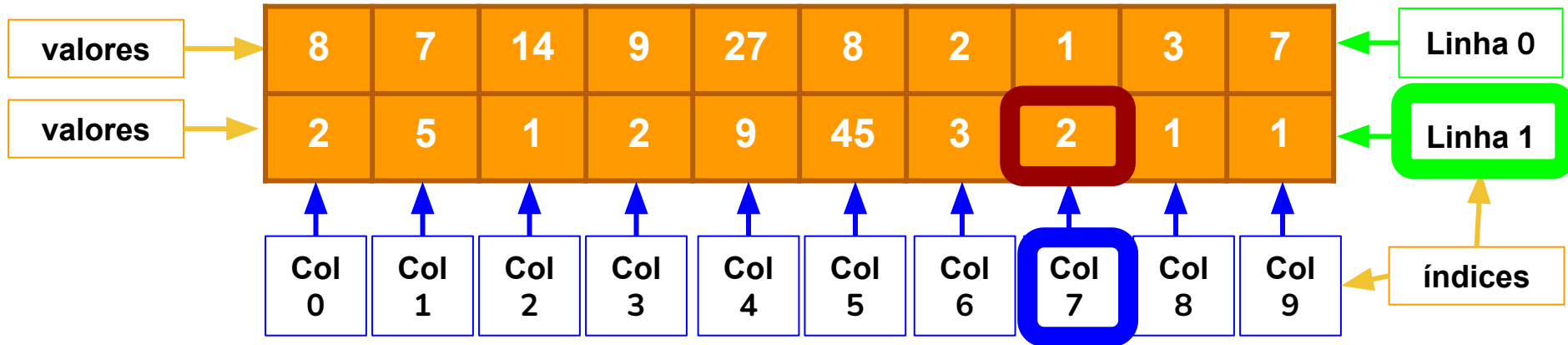
Matrizes

- Necessitam de apenas uma **informação posicional**
 - São **indexados**
- Exemplo: **Qual é o valor no índice - linha 1 e coluna 7?**



Matrizes

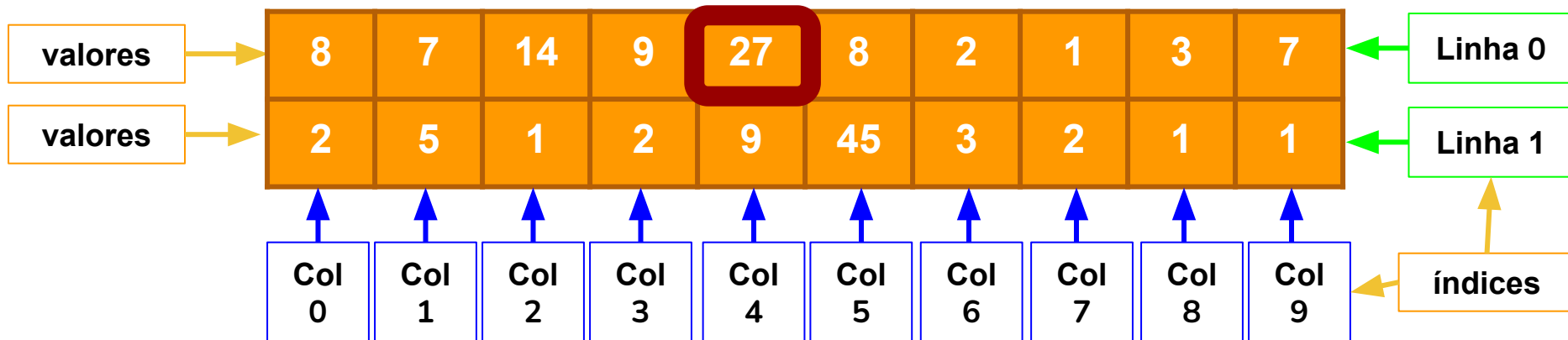
- Necessitam de apenas uma **informação posicional**
 - São **indexados**
- Exemplo: **Qual é o valor no índice - linha 1 e coluna 7?**





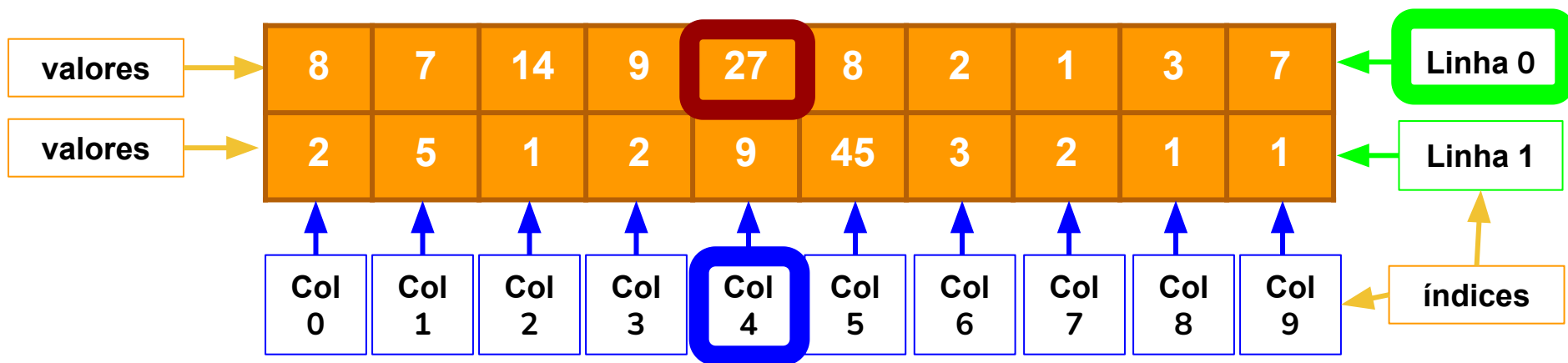
Matrizes

- Necessitam de apenas uma **informação posicional**
 - São **indexados**
- Exemplo: **Qual é o índice do número 27?**



Matrizes

- Necessitam de apenas uma **informação posicional**
 - São **indexados**
- Exemplo: **Qual é o índice do número 27?**





Matrizes

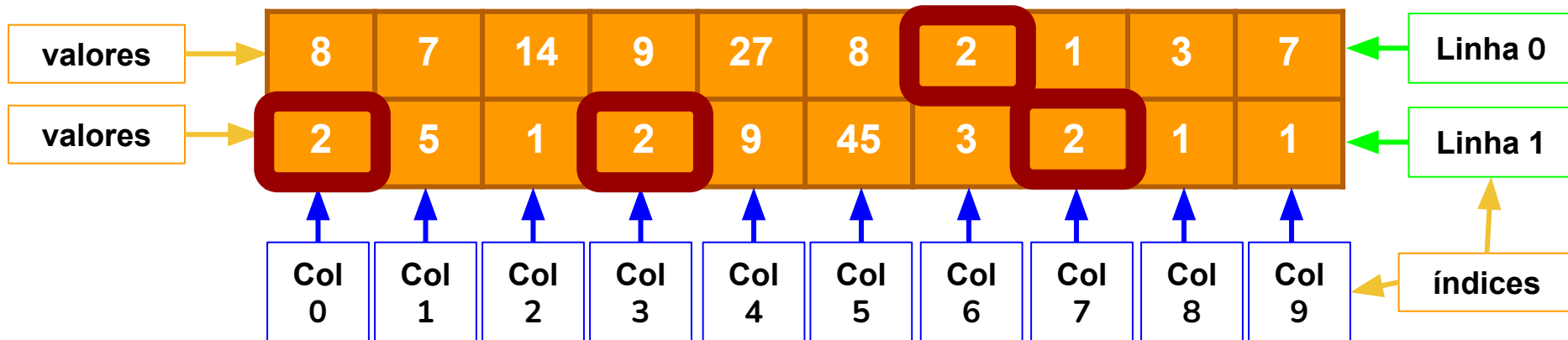
- Necessitam de apenas uma **informação posicional**
 - São **indexados**
- Exemplo: **Qual é o índice do número 2?**

valores	→	8	7	14	9	27	8	2	1	3	7	←	Linha 0
valores	→	2	5	1	2	9	45	3	2	1	1	←	Linha 1
		↑	↑	↑	↑	↑	↑	↑	↑	↑	↑		
		Col 0	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8	Col 9	←	índices



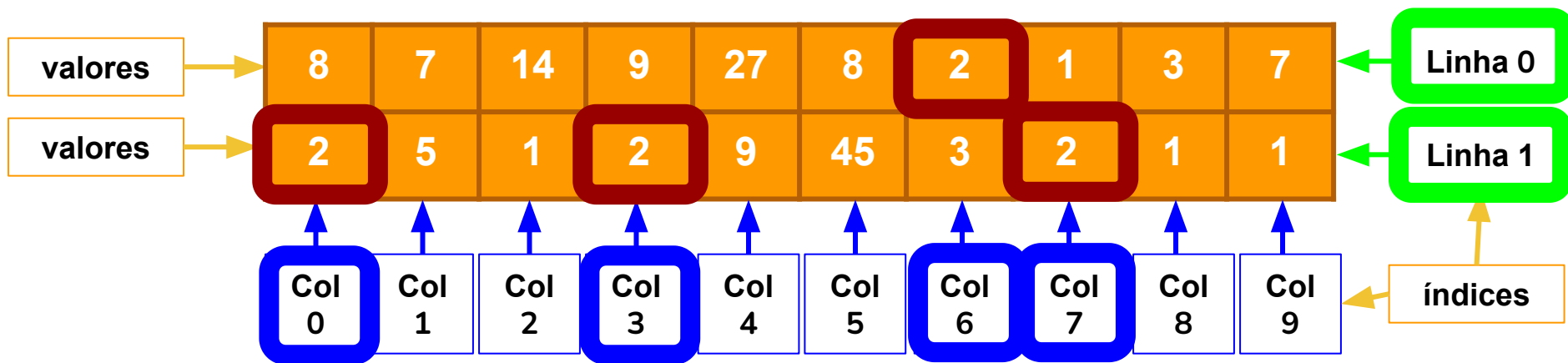
Matrizes

- Necessitam de apenas uma **informação posicional**
 - São **indexados**
- Exemplo: **Qual é o índice do número 2?**



Matrizes

- Necessitam de apenas uma **informação posicional**
 - São **indexados**
- Exemplo: **Qual é o índice do número 2?**





Matrizes

- Vantagens:
 - Mesma vantagem que o vetor:
 - Utilizar **uma única variável** para trabalhar com muitos dados que têm um mesmo significado



Matrizes em C

- Sintaxe:

```
<tipo> nome_matriz [<tam_linha>][<tam_coluna>];
```



Matrizes em C

- Sintaxe:

<tipo> nome_matriz [**<tam_linha>**][**<tam_coluna>**];

- Exemplos:

- **int** v[**100**][**100**]; // matriz de int com 100 linhas e 100 colunas
- **float** f[**8**][**10**]; // matriz de floats com 8 linhas e 10 colunas



Matriz em C

- Considere a matriz v: *int m[2][10];*
- Como atribuir valor a uma posição da matriz *m* ?

m[1][6]

Acesso ao elemento:
linha 1, coluna 6

m =



Matriz em C

- Considere a matriz v: *int m[2][10];*
- Como atribuir valor a uma posição da matriz *m* ?

m[1][6] = 15;

Acesso ao elemento:
linha 1, coluna 6

m =

						15			

Lembre-se que o primeiro elemento tem índice zero

Matriz em C

- Considere a matriz v: *int m[2][10];*
- Como atribuir valor a uma posição da matriz *m* ?

m[0][3] = -7;

Acesso ao elemento:
linha 0, coluna 3

m =

			-7						
						15			

Lembre-se que o primeiro elemento tem índice zero

Matriz em C

- Considere a matriz v: *int m[2][10];*
- Como atribuir valor a uma posição da matriz *m* ?

m[0][0] = 3;

Acesso ao elemento:
linha 0, coluna 0

m =

3			-7						
						15			

Lembre-se que o primeiro elemento tem índice zero

Matriz em C

- Considere a matriz v: *int m[2][10];*
- Como atribuir valor a uma posição da matriz *m* ?

m[1][9] = -11;

Acesso ao elemento:
linha 1, coluna 9

m =

3			-7						
						15			-11

Lembre-se que o primeiro elemento tem índice zero

Matriz em C

- Como acessar um **elemento para leitura**?

```
int i = m[1][9];
```

i vai valer -11

m =

3			-7						
						15			-11



Matrizes em C

- Quando criamos uma matriz, os valores que estão em suas posições são **lixo de memória!**

```
int m[3][5];
```

- Quais são os valores de m?

m =	345546556	-67678	-65478	0	1124
	5656	0	-987	12457899	-987777
	-9787654	-78978976	456456	63233	12587



Matrizes em C

- Para eliminar qualquer resíduo da memória precisamos **inicializar a matriz !**

Matrizes em C

- 3 formas de inicializar!

```
int main(void){
    int main[3][3];
    m[0][0] = 1;
    m[0][1] = 1;
    m[0][2] = 2;
    m[1][0] = 3;
    m[1][1] = 5;
    m[1][2] = 7;
    m[2][0] = 9;
    m[2][1] = 11;
    m[2][2] = 13;
}
```

```
int main(void){
    int mat[][3] = {{1,2,3},{4,5,6},{7,8,9}};
}
```

```
int main(void) {
    int m[3][3];
    int i,j;
    for(i=0;i<3;i++) {
        for(j=0;j<3;j++) {
            m[i][j] = 0;
        }
    }
}
```



Matrizes em C

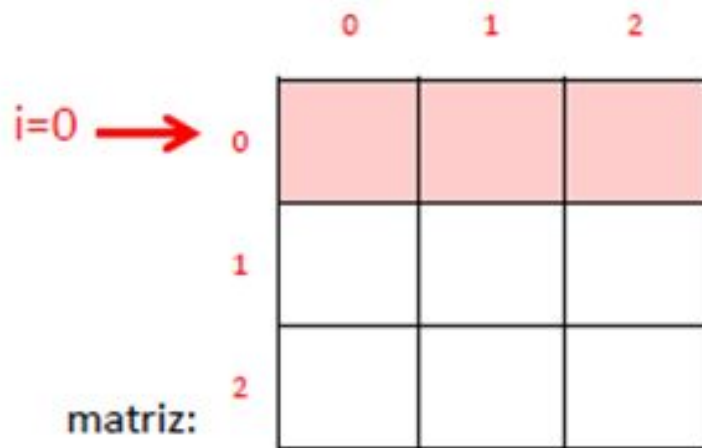
- **Exemplo:** Programa para popular uma matriz 3x3 de inteiros a partir da entrada de dados do usuário



Matrizes em C

- Manipulando matrizes

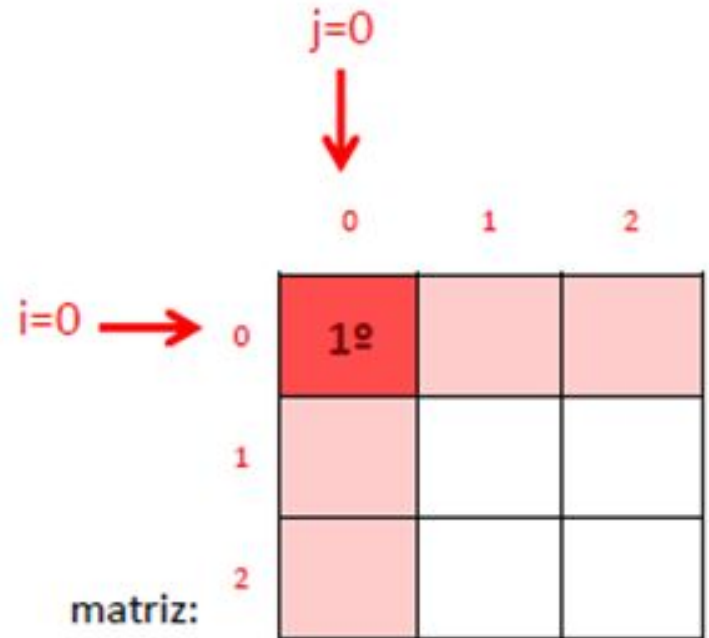
```
int main(void) {  
    int m[3][3];  
    int i,j;  
  
    for(i=0;i<3;i++) {  
        for(j=0;j<3;j++) {  
            m[i][j] = 0;  
        }  
    }  
}
```



Matrizes em C

- Manipulando matrizes

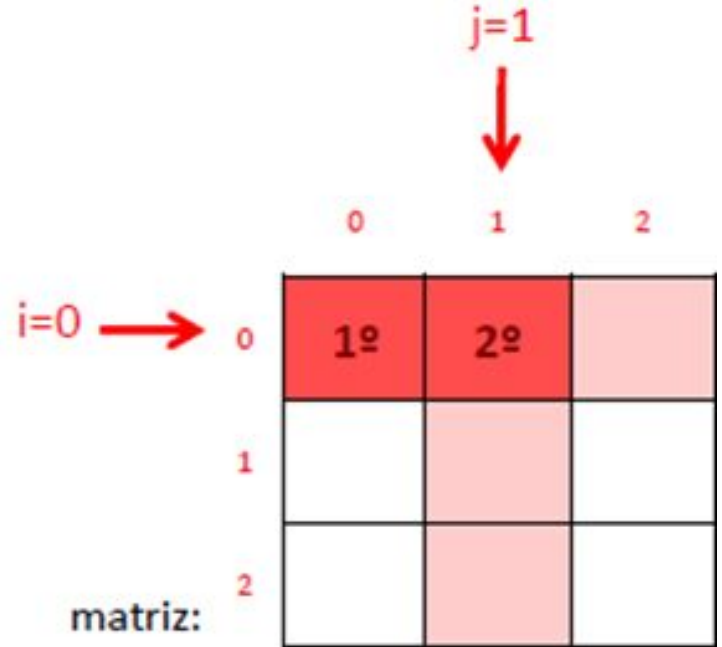
```
int main(void) {  
    int m[3][3];  
    int i,j;  
  
    for(i=0;i<3;i++) {  
        for(j=0;j<3;j++) {  
            m[i][j] = 0;  
        }  
    }  
}
```



Matrizes em C

- Manipulando matrizes

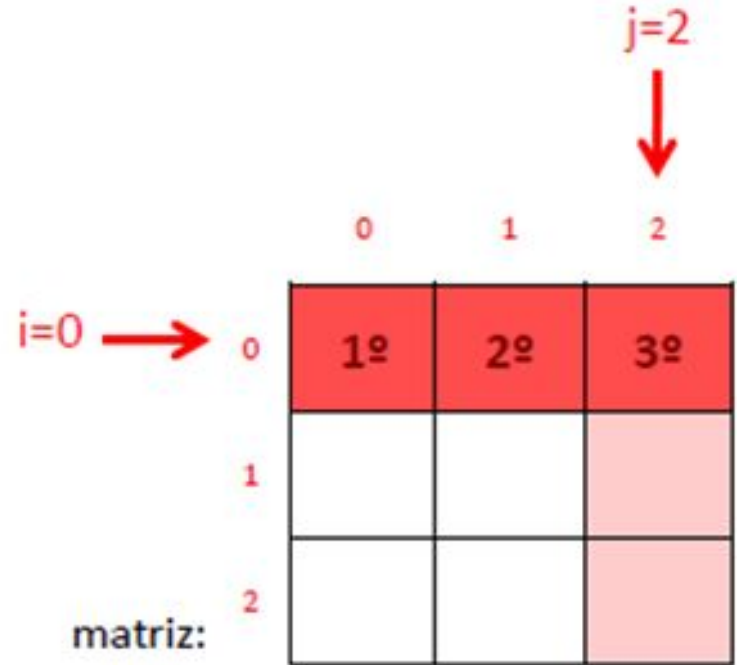
```
int main(void) {  
    int m[3][3];  
    int i,j;  
  
    for(i=0;i<3;i++) {  
        for(j=0;j<3;j++) {  
            m[i][j] = 0;  
        }  
    }  
}
```



Matrizes em C

- Manipulando matrizes

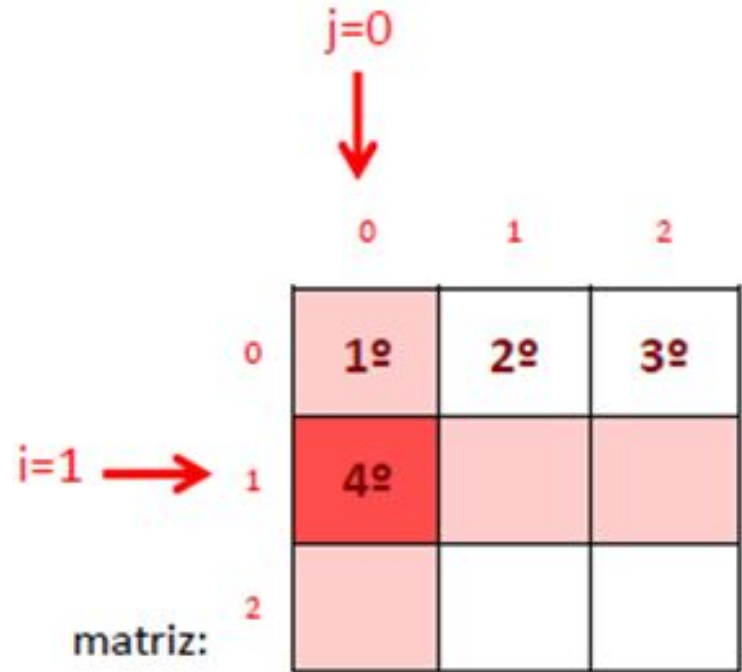
```
int main(void) {  
    int m[3][3];  
    int i,j;  
  
    for(i=0;i<3;i++) {  
        for(j=0;j<3;j++) {  
            m[i][j] = 0;  
        }  
    }  
}
```



Matrizes em C

- Manipulando matrizes

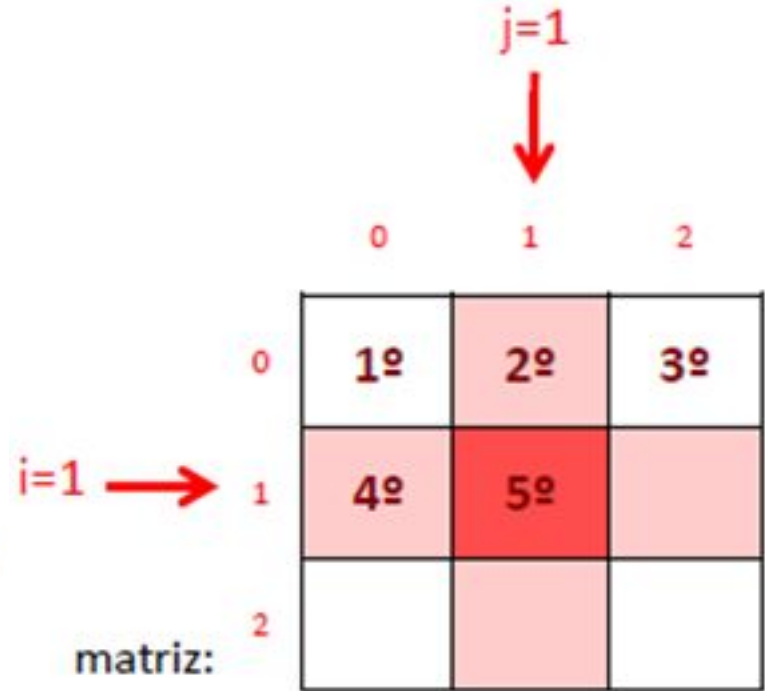
```
int main(void) {  
    int m[3][3];  
    int i,j;  
  
    for(i=0;i<3;i++) {  
        for(j=0;j<3;j++) {  
            m[i][j] = 0;  
        }  
    }  
}
```



Matrizes em C

- Manipulando matrizes

```
int main(void) {  
    int m[3][3];  
    int i,j;  
  
    for(i=0;i<3;i++) {  
        for(j=0;j<3;j++) {  
            m[i][j] = 0;  
        }  
    }  
}
```



Matrizes em C

- Manipulando matrizes

```
int main(void) {  
    int m[3][3];  
    int i,j;  
  
    for(i=0;i<3;i++) {  
        for(j=0;j<3;j++)  
            m[i][j] = 0;  
    }  
}
```

matriz:

	0	1	2
0	1º	2º	3º
1	4º	5º	6º
2			

Diagram description: A 3x3 matrix is shown with indices 0, 1, 2 for both rows and columns. The cell at row 1, column 2 (index [1][2]) is highlighted in red. A red arrow points to this cell from the label 'i=1' on the left, and another red arrow points to it from the label 'j=2' above the column header.

}

Matrizes em C

- Manipulando matrizes

```
int main(void) {  
    int m[3][3];  
    int i,j;  
  
    for(i=0;i<3;i++) {  
        for(j=0;j<3;j++)  
            m[i][j] = 0;  
    }  
}
```

i=2 →
matriz:

	j=0			
	↓			
	0	1	2	
0	1º	2º	3º	
1	4º	5º	6º	
2	7º			

}

Matrizes em C

- Manipulando matrizes

```
int main(void) {  
    int m[3][3];  
    int i,j;  
  
    for(i=0;i<3;i++) {  
        for(j=0;j<3;j++)  
            m[i][j] = 0;  
    }  
}
```

$i=2$ \rightarrow
matriz:

	0	1	2
0	1º	2º	3º
1	4º	5º	6º
2	7º	8º	

}

Matrizes em C

- Manipulando matrizes

```
int main(void) {  
    int m[3][3];  
    int i,j;  
  
    for(i=0;i<3;i++) {  
        for(j=0;j<3;j++)  
            m[i][j] = 0;  
    }  
}
```

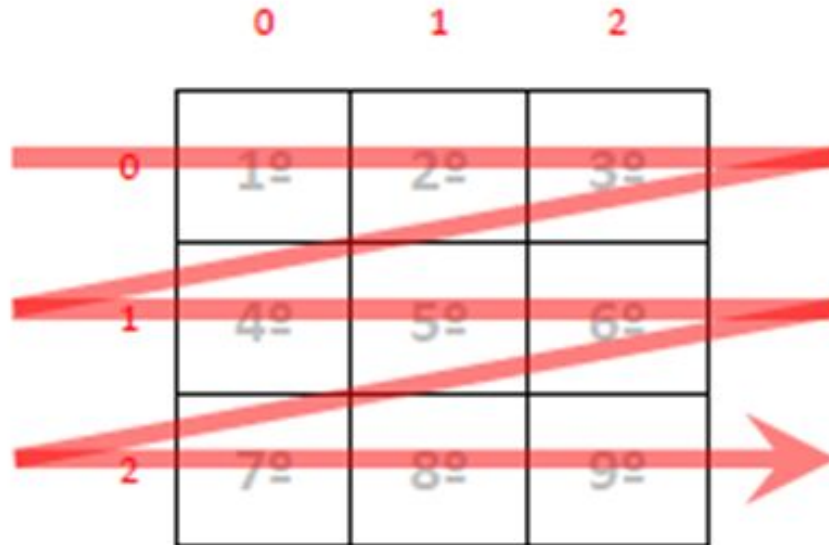
matriz:

	0	1	2
0	1º	2º	3º
1	4º	5º	6º
2	7º	8º	9º



Matrizes em C

- Ordem de visita aos elementos
 - No código anterior a ordem de visitação foi essa:





Matrizes em C - Resumo

Declaração de matrizes:

```
int mat[5][3]; // declara uma matriz de inteiros de 5  
linhas e 3 colunas
```

Preenchimento de “mat” com número 1 em todas as posições:

```
for(i=0; i < 5; i++)  
    for(j=0; j < 3; j++)  
        mat[i][j] = 1;
```


Exercícios

Exercícios



1. Faça um programa que cria uma matriz **m 10 x 15**, sendo que cada elemento é um inteiro gerado aleatoriamente. Então, exiba a matriz completa e, na sequência, somente os elementos da primeira coluna da matriz.
2. Solicitar dados de uma **matriz 4x4** e montar um **vetor de 4 elementos** com a soma dos elementos ímpares de cada linha
3. Faça um programa para receber uma **matriz 3x3** (solicitar ao usuário) e apresentar a **soma dos elementos da diagonal principal** e a matriz na forma como deve ser vista: com linhas e colunas