

Matrizes

Marina Andretta

ICMC-USP

11 de abril de 2016

Uma matriz é uma variável composta bidimensional.

Você pode pensá-la como um vetor em que cada posição armazena um outro vetor.

Para declarar uma matriz, precisamos dizer quantas linhas e quantas colunas a matriz tem e o tipo de dados que ficará armazenado em cada uma de suas posições.

Em pseudo-código, declaramos uma matriz da seguinte forma:

```
<tipo> v[<num.linhas>][<num.colunas>]
```

Em linguagem C:

```
<tipo> v[<num.linhas>][<num.colunas>];
```

Em ambos os casos, <tipo> é o tipo de dado que será armazenado em cada posição da matriz, <num.linhas> é o número de linhas da matriz e <num.colunas> é o número de colunas da matriz.

Exemplos de declaração de matrizes, em pseudo-código, são:

```
inteiro mat[3][5]
```

```
real mat[2][7]
```

```
caracter mat[10][3]
```

```
logico mat[100][89]
```

Em linguagem C, as mesmas matrizes são declarados da seguinte forma:

```
int mat[3][5];
```

```
float mat[2][7];
```

```
char mat[10][3];
```

```
int mat[100][89];
```

Quando declaramos uma matriz usando

```
inteiro mat [3] [5]
```

em pseudo-código ou

```
int mat [3] [5] ;
```

em linguagem C, o que estamos fazendo é alocar na memória 15 espaços para variáveis inteiras, um seguido do outro. Na memória, primeiramente ficam os elementos da primeira linhas, depois os da segunda e assim por diante.

Cada um destes espaços será acessado usando um índice correspondente à linha e outro correspondente à coluna da matriz.

	0	1	2	3	4
0					
1					
2					

Matrizes

Por exemplo, se queremos guardar o valor 7 na posição 1×3 da matriz `mat`, escrevemos, em pseudo-código,

```
mat[1][3] ← 7
```

Em linguagem C, escrevemos

```
mat[1][3] = 7;
```

O efeito disso é

	0	1	2	3	4
0					
1				7	
2					

Exemplo de um algoritmo completo: em pseudo-código

```
programa matriz
  declaracoes
    real matriz[2][3]
    inteiro i,j
inicio
  escreva(“Digite os elementos da matriz: ”)
  para i ← 0 ate 1 passo 1
    para j ← 0 ate 2 passo 1
      leia(matriz[i][j])
    fimpara
  fimpara
  para i ← 0 ate 1 passo 1
    para j ← 0 ate 2 passo 1
      escreva(matriz[i][j], “ ”)
    fimpara
    escreva(“\n”)
  fimpara
fim
```


Vamos simular a execução do algoritmo, supondo que o usuário digitou os números 1, 2, 3, 4, 5 e 6. Depois da declaração das variáveis, temos

`i =`

`j =`

```
matriz
```

	0	1	2
0			
1			

O primeiro laço se inicia, com o contador i começando com valor 0.

$i = 0$

$j =$

		0	1	2
matriz	0			
	1			

O segundo laço se inicia, com o contador j começando com valor 0.

$i = 0$

$j = 0$

		0	1	2
matriz	0			
	1			

Depois de lido o primeiro valor (1), temos

`i = 0`

`j = 0 1`

		0	1	2
matriz	0	1		
	1			

Depois de lido o segundo valor (2), temos

`i = 0`

`j = 0 1 2`

		0	1	2
matriz	0	1	2	
	1			

Depois de lido o terceiro valor (3), temos

`i = 0`

`j = 0 1 2 3`

		0	1	2
matriz	0	1	2	3
	1			

Neste ponto, o laço interno se encerra e continuamos a execução do segundo laço. Incrementamos em 1 o contador i :

$i =$ ~~0~~ 1

$j =$ ~~0~~ ~~1~~ ~~2~~ 3

		0	1	2
matriz	0	1	2	3
	1			

O segundo laço se inicia novamente, fazendo com que o contador j seja definido como 0:

$i =$ ~~0~~ 1

$j =$ ~~0~~ ~~1~~ ~~2~~ ~~3~~ 0

		0	1	2
matriz	0	1	2	3
	1			

Depois de lido o quarto valor (4), temos

i = ~~0~~ 1

j = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ 1

		0	1	2
matriz	0	1	2	3
	1	4		

Depois de lido o quinto valor (5), temos

i = ~~0~~ 1

j = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ 2

		0	1	2
matriz	0	1	2	3
	1	4	5	

Depois de lido o sexto valor (6), temos

i = ~~0~~ 1

j = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ 3

		0	1	2
matriz	0	1	2	3
	1	4	5	6

Neste ponto, o laço interno se encerra o laço externo continua a ser executado. O contador i é incrementado de 1.

$i =$ ~~0~~ ~~1~~ 2

$j =$ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ 3

		0	1	2
matriz	0	1	2	3
	1	4	5	6

Neste ponto, o laço externo também se encerra. Então, temos o seguinte:

i = ~~0~~ ~~1~~ ~~2~~ 0

j = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ 0

matriz

	0	1	2
0	1	2	3
1	4	5	6

Tela:

1

Neste ponto, o laço externo também se encerra. Então, temos o seguinte:

i = ~~0~~ ~~1~~ ~~2~~ 0

j = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ 1

matriz

	0	1	2
0	1	2	3
1	4	5	6

Tela:

1 2

Neste ponto, o laço externo também se encerra. Então, temos o seguinte:

i = ~~0~~ ~~1~~ ~~2~~ 0

j = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ 2

matriz

	0	1	2
0	1	2	3
1	4	5	6

Tela:

1 2 3

Teste de mesa

Neste ponto, o laço externo também se encerra. Então, temos o seguinte:

i = ~~0~~ ~~1~~ ~~2~~ 0

j = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ 3

matriz

	0	1	2
0	1	2	3
1	4	5	6

Tela:

1 2 3

Neste ponto, o laço externo também se encerra. Então, temos o seguinte:

i = ~~0~~ ~~1~~ ~~2~~ ~~0~~ 1

j = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ 0

matriz

	0	1	2
0	1	2	3
1	4	5	6

Tela:

1 2 3

4

Neste ponto, o laço externo também se encerra. Então, temos o seguinte:

i = ~~0~~ ~~1~~ ~~2~~ ~~0~~ 1

j = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ 1

matriz

	0	1	2
0	1	2	3
1	4	5	6

Tela:

1 2 3

4 5

Teste de mesa

Neste ponto, o laço externo também se encerra. Então, temos o seguinte:

i = ~~0~~ ~~1~~ ~~2~~ ~~0~~ 1

j = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~0~~ ~~1~~

2

matriz

	0	1	2
0	1	2	3
1	4	5	6

Tela:

1 2 3

4 5 6

Teste de mesa

Neste ponto, o laço externo também se encerra. Então, temos o seguinte:

```
i = 0 1 2 0 1
j = 0 1 2 3 0 1 2 3 0 1 2 3 0 1
2 3
```

```
matriz  0  1  2
0  

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
|---|---|---|


1  

|   |   |   |
|---|---|---|
| 4 | 5 | 6 |
|---|---|---|


```

Tela:

```
1 2 3
4 5 6
```

Neste ponto, o laço externo também se encerra. Então, temos o seguinte:

```
i = 0 1 2 0 1 2
j = 0 1 2 3 0 1 2 3 0 1 2 3 0 1
2 3
```

```
matriz  0  1  2
0  

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
|---|---|---|


1  

|   |   |   |
|---|---|---|
| 4 | 5 | 6 |
|---|---|---|


```

Tela:

```
1 2 3
4 5 6
```

Exemplo do mesmo algoritmo completo: em linguagem C

```
int main () {
    float matriz[2][3];
    int i,j;

    printf("Digite os elementos da matriz:  ");
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 3; j++) {
            scanf("%f", &matriz[i][j]);
        }
    }
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 3; j++) {
            printf("%f  ", matriz[i][j]);
        }
        printf("\n");
    }
    return(0);
}
```