

Algoritmos e Lógica de Programação

Matrizes

Profa. Eliane Oliveira Santiago

Matriz

Uma matriz é um conjunto de informações de um mesmo tipo de dados, podendo ser organizada com múltiplas dimensões.

Note que matriz é um nome particular dado à matriz multidimensional, seja ela de dimensão dois ou mais. Por exemplo, uma matriz bidimensional pode conter informações dispostas em duas coordenadas x e y , ou seja, linha e coluna; uma matriz tridimensional pode conter informações dispostas em três coordenadas x , y e z , ou seja, linha, coluna e altura e assim por diante. Neste capítulo, nos preocuparemos em estudar as matrizes bidimensionais.

Exemplo de matriz

Mat	0	1	2	3	4
0	carteira1	carteira2	carteira3	carteira4	carteira5
1	carteira6	carteira7	carteira8	carteira9	carteira10
2	carteira11	carteira12	carteira13	carteira14	carteira15
3	carteira16	carteira17	carteira18	carteira19	carteira20
4	carteira21	carteira22	carteira23	carteira24	carteira25
5	carteira26	carteira27	carteira28	carteira29	carteira30
6	carteira31	carteira32	carteira33	carteira34	carteira35
7	carteira36	carteira37	carteira38	carteira39	carteira40
8	carteira41	carteira42	carteira43	carteira44	carteira45
9	carteira46	carteira47	carteira48	carteira49	carteira50

Neste exemplo, possuímos uma matriz chamada **Mat**, com dimensão, ou tamanho da matriz, igual a 10 por 5, ou seja, 10 linhas e 5 colunas, com cinquenta posições: 0x0 , 0x1 , 0x2 , 0x3 , 0x4 , 1x0 , 1x1 , 1x2 , 1x3 , 1x4 , 2x0 , 2x1 , 2x2 , 2x3 , 2x4 , 3x0 , 3x1 , 3x2 , 3x3 , 3x4 , 4x0 , 4x1 , 4x2 , 4x3 , 4x4 , 5x0 , 5x1 , 5x2 , 5x3 , 5x4 , 6x0 , 6x1 , 6x2 , 6x3 , 6x4 , 7x0 , 7x1 , 7x2 , 7x3 , 7x4 , 8x0 , 8x1 , 8x2 , 8x3 , 8x4 , 9x0 , 9x1 , 9x2 , 9x3 e 9x4. Na posição 0x0 da matriz **Mat**, temos a carteira1; na posição 0x1 da matriz **Mat**, temos a carteira2; até que na posição 9x4 da matriz **Mat**, temos a carteira50.

Declaração de matriz

Sintaxe:

<nome da matriz> [<nro de linhas>] [<nro de colunas>] <tipo dos elementos matriz>;

Exemplo

Var

```
Mat_exemplo [0..9][0..4] : inteiro;
```

Neste exemplo, declaramos uma matriz chamada **Mat_exemplo** com 10 linhas e 5 colunas totalizando 50 elementos do tipo inteiro.

Exemplo de matriz

notas		1º Bim	2º Bim	3º Bim	4º Bim	Média Final
	posições	0	1	2	3	4
Aluno 1	0	9.0	8.0	8.5	9.0	8.5
Aluno 2	1	5.0	6.0	7.0	8.0	6.5
Aluno 3	2	7.5	7.5	7.5	7.5	7.5
Aluno 4	3	5.0	9.0	3.0	7.0	6.0
Aluno 5	4	8.0	6.0	6.0	7.0	7.0
Aluno 6	5	9.0	8.0	9.0	10.0	9.0
Aluno 7	6	8.0	8.0	8.0	8.0	8.0
Aluno 8	7	8.5	6.0	7.5	6.0	7.0
Aluno 9	8	9.5	8.5	8.5	9.5	9.0
Aluno 10	9	5.0	6.5	7.0	7.5	6.5

- **notas** é o nome da matriz;

a matriz **notas** possui 10 linhas e 5 colunas, totalizando 50 informações.

Por isso, tem tamanho 10x5, ou seja, 50 posições: de 0x0 até 9x4;

- a posição 0x0 da matriz, notas[0][0], possui a nota 9.0;

- a posição 0x1 da matriz, notas[0][1], possui a nota 8.0; e assim por diante.

notas		1º Bim	2º Bim	3º Bim	4º Bim	Média Final
	posições	0	1	2	3	4
Aluno 1	0	9.0	8.0	8.5	9.0	8.5

```
int x = 10;
```

```
int valor = 5;
```

```
x = x - 1;                      ⇔        x -= 1;
```

```
x = x + 5;                      ⇔        x += 5;
```

```
valor = valor * x; ⇔            valor *= x;
```

```
notas[linha][4] = notas[linha][4] + notas[linha][coluna];
```

```
notas[linha][4] += notas[linha][coluna];
```

Atribuição

Sintaxe:

<nome da matriz> [<nro. da linha>] [<nro. da coluna>] ← <valor>;

Exemplos:

notas [0][0] ← 9.0;

notas [2] ← {{ 9.0 , 10.0 , 8.5, 9.0, 10.0}} : **real;**

Neste exemplo, declaramos uma matriz chamada **notas** de tamanho 2x3, ou seja, de 6 posições, de forma que a posição 0x0 da matriz possua o valor 9.0, a posição 0x1 da matriz possua o valor 10.0, a posição 0x2 da matriz possua o valor 8.5, a posição 1x0 da matriz possua o valor 5.5, a posição 1x1 da matriz possua o valor 5.0 e a posição 1x2 da matriz possua o valor 7.5.

Mostrar os elementos de uma matriz

Sintaxe:

escreva(<nome da matriz>[<nro. da linha>][<nro. da coluna>]);

escreva(notas [0][0])

escreva("A nota é: ", notas[0][0])

Exemplo

Algoritmo Mostrar

var

Mat [0..4][0..4] , i : inteiro

Inicio

para i de 0 até 4 **passo** + 1 **faça**

para j de 0 até 4 **passo** + 1 **faça**

escreva ("Digite um valor inteiro"); // mensagem ao usuário

leia(Mat[i][j]); // entrada de dados

escreva(Mat[i][j]); // saída de resultados

fim_para;

fim_para;

fim_algoritmo.

	0	1	2	coluna
0	0	1	2	
1	1	2	3	
2	2	3	4	
3				
4				

linha

Linha	coluna	M1[linha,coluna]
0	0	M1[0][0] ← 0
	1	M1[0][1] ← 1
	2	M1[0][2] ← 2
1	0	M1[1][0] ← 1
	1	M1[1][1] ← 2
	2	M1[1][2] ← 3
2	0	M1[2][0] ← 2
	1	M1[2][1] ← 3
	2	M1[2][2] ← 4

```

for(int linha=0; linha<5; linha++)
    for(int coluna=0; coluna<3; coluna++)
        M1[linha][coluna] ← linha+coluna
    }
}

```

1B 2B MS

0 1 2

colunas

Aluno 1

0

10.0

5.0

7.5

Aluno 2

1

7.5

9.0

8.25

Aluno 3

2

8.0

8.5

8.25

Aluno 4

3

10.0

6.0

8.0

Aluno 5

linh
as

7.5

4.5

5.5

Linha	Coluna	Notas[L,C]
0	0	$\text{Notas}[0,0] \leftarrow 10.0$
	1	$\text{Notas}[0,1] \leftarrow 5.0$
		$\text{Notas}[0,2] \leftarrow (\text{notas}[0,0] + \text{notas}[0,1]) / 2$
1	0	$\text{Notas}[1,0] \leftarrow 7.5$
	1	$\text{Notas}[1,1] \leftarrow 9.0$
		$\text{Notas}[1,2] \leftarrow (\text{notas}[1,0] + \text{notas}[1,1]) / 2$
2		

Exercícios com matrizes

1. Desenvolva um algoritmo que receba e mostre 120 valores alfanuméricos numa matriz 10x12.

Exercícios com matrizes

2. Desenvolva um algoritmo que receba 20 valores numéricos inteiros numa matriz A de dimensões 10x2 e 20 valores numéricos inteiros numa matriz B de dimensões 10x2. Construa uma matriz C com dimensões 10x2, onde cada posição possua a soma dos elementos das matrizes A e B em suas respectivas posições. Mostre os elementos das três matrizes.

A	0	1
0	1	2
1	9	8
2	10	15
3	25	30
4	15	10

B	0	1
0	5	6
1	1	2
2	20	30
3	2	45
4	5	5

C	0	1
0	6	8
1	10	10
2	30	45
3	27	75
4	20	15

Exercícios com matrizes

3. Desenvolva um algoritmo que receba 50 valores reais numa matriz X de dimensões 5×10 e 30 valores reais numa matriz Y de dimensões 3×10 . Construa uma matriz Z de dimensões 8×10 com a concatenação das matrizes X e Y , ou seja, os elementos das 5 primeiras linhas são os mesmos elementos da matriz X e os elementos das três últimas linhas são os mesmos da matriz Y . Mostre os elementos das três matrizes.

Desenvolva um algoritmo que receba 50 valores reais numa matriz X de dimensões 5x10 e 30 valores reais numa matriz Y de dimensões 3x10. Construa uma matriz Z de dimensões 8x10 com a concatenação das matrizes X e Y, ou seja, os elementos das 5 primeiras linhas são os mesmos elementos da matriz X e os elementos das três últimas linhas são os mesmos da matriz Y. Mostre os elementos das três matrizes.

X	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	11	12	13	14	15	16	17	18	19	20
2	21	22	23	24	25	26	27	28	29	30
3	31	32	33	34	35	36	37	38	39	40
4	41	42	43	44	45	46	47	48	49	50

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

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

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

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

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

Lin	Col	Z[lin,col]
0	0 : 10	$Z[0,0] \leftarrow X[0,0]$ $Z[5,0] \leftarrow Y[0,0]$
1	0 : 10	
2	0 : 10	

Para lin de 0 ate 7 passo 1 faca
 para col de 0 ate 9 passo 1 faca
 se(lin<3) entao
 $Z[\text{lin},\text{col}] \leftarrow X[\text{lin},\text{col}]$
 $Z[\text{lin}+5,\text{col}] \leftarrow Y[\text{lin},\text{col}]$
 senao
 $Z[\text{lin},\text{col}] \leftarrow X[\text{lin},\text{col}]$
 fimse
 fimpara
 fimpara

Exercícios com matrizes

4. Desenvolva um algoritmo que receba 50 valores numéricos reais numa matriz 10×5 , calcule e armazene numa segunda matriz os 50 valores da primeira matriz multiplicados por 7. Mostre os valores das duas matrizes.

Exercícios com matrizes

5. Desenvolva um algoritmo que calcule a tabuada dos números de 1 a 10 e armazene numa matriz de dimensões 10×10 o resultado da tabuada. Mostre os elementos da matriz. Na primeira coluna da matriz, armazene a tabuada do número 1, na segunda coluna da matriz, armazene a tabuada do número 2 e assim por diante, até a última coluna da matriz, armazene a tabuada do número 10.
6. Desenvolva um algoritmo que receba 49 valores numéricos inteiros numa matriz de dimensões 7×7 , calcule e mostre os números pares, suas posições e a soma dos números pares, bem como os números ímpares, suas posições e a quantidade de números ímpares.

Exercícios com matrizes (4)

7. Desenvolva um algoritmo que receba 15 valores inteiros numa matriz X de dimensões 3×5 , receba 15 valores inteiros numa matriz Y de dimensões 3×5 e concatene alternadamente os elementos das matrizes X e Y numa terceira matriz Z de dimensões 3×10 .

Os elementos das colunas pares da matriz Z são os mesmos da matriz X e os elementos das colunas ímpares da matriz Z são os mesmos da matriz Y . Mostre os elementos das três matrizes.

Exercícios com matrizes (4)

7. Desenvolva um algoritmo que receba 15 valores inteiros numa matriz X de dimensões 3x5, receba 15 valores inteiros numa matriz Y de dimensões 3x5 e concatene alternadamente os elementos das matrizes X e Y numa terceira matriz Z de dimensões 3x10. Os elementos das colunas pares da matriz Z são os mesmos da matriz X e os elementos das colunas ímpares da matriz Z são os mesmos da matriz Y. Mostre os elementos das três matrizes.

X	0	1	2	3	4
0	0	2	4	6	8
1					
2					

Y	0	1	2	3	4
0	1	3	5	7	9
1	11				
2					

Z	0	1	2	3	4	5	6	7	8	9
0		Y	X	Y	X	Y	X	Y	X	Y
1	X	Y	X	Y	X	Y	X	Y	X	Y
2	X	Y	X	Y	X	Y	X	Y	X	Y

Exercícios com matrizes (5)

8. Desenvolva um algoritmo que receba 75 valores numa matriz de dimensões 15×5 , armazene estes valores numa segunda matriz de dimensões 15×5 a partir do centro para a primeira e a última posições e de modo alternado, ou seja, o primeiro elemento da primeira matriz estará na posição do meio da segunda matriz, o segundo elemento da primeira matriz estará na posição à esquerda da posição do meio da segunda matriz, o terceiro elemento da primeira matriz estará na posição à direita da posição do meio da segunda matriz e assim por diante. Mostre os elementos das duas matrizes.

Days 1 - 10
Teach yourself variables, constants, arrays, strings, expressions, statements, functions,...



Days 11 - 21
Teach yourself program flow, pointers, references, classes, objects, inheritance, polymorphism,



Days 22 - 697
Do a lot of recreational programming. Have fun hacking but remember to learn from your mistakes.



Days 698 - 3648
Interact with other programmers. Work on programming projects together. Learn from them.



Days 3649 - 7781
Teach yourself advanced theoretical physics and formulate a consistent theory of quantum gravity.



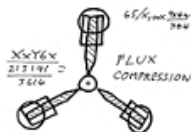
Days 7782 - 14611
Teach yourself biochemistry, molecular biology, genetics,...



Day 14611
Use knowledge of biology to make an age-reversing potion.



Day 14611
Use knowledge of physics to build flux capacitor and go back in time to day 21.



Day 21
Replace younger self.



As far as I know, this
is the easiest way to
"Teach Yourself C++ in 21 Days".

Bibliografias

BÁSICA

- GOMES, Ana Fernanda A. Campos, Edilene Aparecida V. Fundamentos da Programação de Computadores – Algoritmos, Pascal e C/C++. Prentice Hall, 2007.
- CARBONI, Irenice de Fátima. Lógica de Programação. Thomson.
- XAVIER, Gley Fabiano Cardoso. Lógica de Programação - Cd-rom. Senac São Paulo – 2007.

COMPLEMENTAR

- FORBELLONE, André Luiz Villar. Eberspache, Henri Frederico. Lógica de Programação – A construção de Algoritmos e Estrutura de Dados. Makron Books, 2005.
- LEITE, Mário - Técnicas de Programação – Brasport - 2006.
- PAIVA, Severino – Introdução à Programação – Ed. Ciência Moderna – 2008.
- PAULA, Everaldo Antonio de. SILVA, Camila Ceccatto da. Lógica de Programação –Viena – 2007.
- CARVALHO, Fábio Romeu, ABE, Jair Minoru. Tomadas de decisão com ferramentas da lógica paraconsistente anotada: Método Paraconsistente de Decisão (MPD), Editora Edgard Blucher Ltda. - 2012.