

Lógica de programação

Matrizes

Walisson Pereira

walisson_pereira@uvanet.br

Universidade Estadual Vale do Acaraú

Matrizes

Exercícios

Referências

Matrizes

Uma **Matriz** é uma variável composta homogênea multidimensional. Ela é formada por uma sequência de variáveis, todas do mesmo tipo, com o mesmo identificador (mesmo nome), e alocadas sequencialmente na memória. Uma vez que as variáveis têm o mesmo nome, o que as distingue são índices que referenciam sua localização dentro da estrutura. Uma variável do tipo matriz precisa de um índice para cada uma de suas dimensões.

Como criar uma matriz em C:

Preenchendo uma matriz 5×5 preenchida com zeros.

```
1  int m[5][5];  
2  int i,j;  
3  for (i = 0; i < 5; i++)  
4      for (j = 0; j < 5; j++)  
5          m[i][j] = 0;
```

Uma matriz 5×5 preenchida com zeros.

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

Como criar uma matriz:

Preenchendo uma matriz 5×5 com valores de 1 a 25.

```
1 int i, j, valor = 1
2 int m[5][5];
3 for (i = 0; i < 5; i++)
4     for (j = 0; j < 5; j++)
5     {
6         m[i][j] = valor;
7         valor++;
8     }
```

Uma matriz 5×5 com valores de 1 a 25.

		coluna				
		0	1	2	3	4
linha	0	1	2	3	4	5
	1	6	7	8	9	10
	2	11	12	13	14	15
	3	16	17	18	19	20
	4	21	22	23	24	25

Acessamos um valor da matriz indicando o seu índice.

```
1 printf("%d", m[1][2]);  
2 printf("%d", m[0][1]);  
3 x = m[3][2];
```

Alteramos um valor da matriz atribuindo valor ao índice especificado.

```
1 m[1][2] = 33;
```

Para imprimir na tela uma matriz 5×5 :

```
1 int i,j;
2 for (i = 0; i < 5; i++) {
3     for (j = 0; j < 5; j++) {
4         printf("%d ", mat[i][j]);
5     }
6     printf("\n");
7 }
```

Exemplo de leitura e impressão de uma matriz 3×3 :

```
1  #include <stdio.h>
2
3  int main () {
4      int i, j, mat[3][3];
5      for (i = 0; i < 3; i++)
6          for (j = 0; j < 3; j++)
7              scanf("%d", &mat[i][j]);
8      for (i = 0; i < 3; i++) {
9          for (j = 0; j < 3; j++) {
10             printf("%d ", mat[i][j]);
11         }
12         printf("\n");
13     }
14 }
```

Em algumas situações é desejável gerar **números aleatórios**, seja para testar um algoritmo, seja para outra funcionalidade.

Usando números pseudo-aleatórios

Em C podemos usar o comando **rand()** para gerar números aleatórios.

Exemplo:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main () {
5     int numero;
6     numero = rand();
7     printf("%d\n", numero);
8 }
```

Terminal:

```
1 1804289383
```

Matrizes

Nem sempre desejamos um número gerado muito grande, então podemos usar o módulo para diminuir o seu escopo.

Exemplo: gerando um número entre 0 e 9.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main () {
5      int numero;
6      numero = rand() % 10;
7      printf("%d\n", numero);
8  }
```

Terminal:

```
1  3
```

Usando números pseudo-aleatórios

Exemplo: gerando uma matriz 5×5 de números inteiros aleatórios.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main () {
5      int mat[5][5];
6      int i,j;
7      for (i = 0; i < 5; i++)
8          for (j = 0; j < 5; j++)
9              mat[i][j] = rand() % 10;
10     for (i = 0; i < 5; i++) {
11         for (j = 0; j < 5; j++)
12             printf("%d ", mat[i][j]);
13         printf("\n");
14     }
15 }
```

Para testar um algoritmo, é interessante que sempre seja gerado o mesmo conjunto de números aleatórios. Desta forma, é mais fácil identificar erros.

Por padrão, **rand()** sempre entregará a mesma sequência de números.

Podemos alterar essa sequência, alterando a **semente** da função **rand()** a partir do comando **srand(semente)**.

Usando números pseudo-aleatórios

Exemplo: gerando uma matriz 5×5 de números inteiros aleatórios, usando o número 10 como semente.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main () {
4     srand(10);
5     int mat[5][5];
6     int i,j;
7     for (i = 0; i < 5; i++)
8         for (j = 0; j < 5; j++)
9             mat[i][j] = rand() % 10;
10    for (i = 0; i < 5; i++) {
11        for (j = 0; j < 5; j++)
12            printf("%d ", mat[i][j]);
13        printf("\n");
14    }
15 }
```

Podemos pesquisar se um elemento está ou não em uma matriz, verificando do primeiro ao último elemento se o valor procurado estiver presente:

No código a seguir, criamos uma matriz preenchida com a função **rand()** e depois é lido um valor inteiro e retornado se o valor foi encontrado ou não na matriz.

Pesquisa

```
1  int mat[5][5];
2  int i,j;
3  for (i = 0; i < 5; i++)
4      for (j = 0; j < 5; j++)
5          mat[i][j] = rand() % 10;
6  int achou = 0, pesquisar;
7  scanf("%d", &pesquisar);
8  for (i = 0; i < 5; i++)
9      for (j = 0; j < 5; j++)
10         if (mat[i][j] == pesquisar) {
11             achou = 1;
12             break;
13         }
14  if (achou)
15     printf("%d foi encontrado\n", pesquisar);
16  else
17     printf("%d nao encontrado\n", pesquisar);
```

Exercícios

Exercícios

1. Faça um programa que preencha uma matriz 3×5 com números inteiros entre 1 e 20, calcule e mostre a quantidade de elementos entre 15 e 20.
2. Elabore um programa que preencha uma matriz 6×3 , calcule e mostre:
 - o maior elemento da matriz e sua respectiva posição, ou seja, linha e coluna;
 - o menor elemento da matriz e sua respectiva posição, ou seja, linha e coluna.
3. Elabore um programa que preencha uma matriz 10×10 com números inteiros aleatórios, execute as trocas especificadas a seguir e mostre o antes e depois:
 - a linha 2 com a linha 8;
 - a coluna 4 com a coluna 9;
 - a diagonal principal com a diagonal secundária;
 - a linha 5 com a coluna 9.

Referências

- 1 ASCENCIO, A.F.G.; CAMPOS, E.A.V. de. Fundamentos da Programação de Computadores. Algoritmos, Pascal e C/C++. São Paulo: Pearson Prentice Hall, 2012.
- 2 VAREJÃO, F. M. V. Introdução à programação: uma nova abordagem usando C. Rio de Janeiro: Elsevier, 2015.
- 3 BACKES, A. Linguagem C: completa e descomplicada. Rio de Janeiro: Elsevier, 2013.