

Arquitetura de Computadores, Algoritmos, Lógica de Programação e Circuitos Digitais

Danilo Rios
22/08/2025

Recapitulando

Array

- O que é?
 - Vetor
 - É uma variável que possui vários valores

Atenção!

- A primeira posição do array é a posição ZERO!
 - Não é a posição 1

Array

- Vetor é uma variável unidimensional
- Podemos fazer um array bidimensional -> Matriz
- Podemos fazer um array tridimensional -> Cubo
- Podemos fazer um array com 4, 5, 6 ... dimensões
 - Não temos um objeto no mundo real, mas funciona

Array

- Na aula de hoje foco é no array com mais de 1 dimensão

Array bidimensional

- É uma “junção” de vários arrays de 1 dimensão

L = 0, C = 0	L = 0, C = 1	L = 0, C = 2	L = 0, C = 3	L = 0, C = 4
L = 1, C = 0	L = 1, C = 1	L = 1, C = 2	L = 1, C = 3	L = 1, C = 4
L = 2, C = 0	L = 2, C = 1	L = 2, C = 2	L = 2, C = 3	L = 2, C = 4
L = 3, C = 0	L = 3, C = 1	L = 3, C = 2	L = 3, C = 3	L = 3, C = 4
L = 4, C = 0	L = 4, C = 1	L = 4, C = 2	L = 4, C = 3	L = 4, C = 4

Array bidimensional

- Matriz com 2 dimensões
- Como é feita a declaração?

Array bidimensional

- Matriz com 2 dimensões
- Como é feita a declaração?
 - `[]` é utilizado para determinar o tamanho de 1 dimensão
 - Para ter 2 dimensões teremos `[] []`

Array bidimensional

- Matriz com 2 dimensões
- Como percorremos todas as posições?

Array bidimensional

- Matriz com 2 dimensões
- Como percorremos todas as posições?
 - Precisamos de 1 loop para percorrer 1 dimensão
 - Precisamos de 1 loop dentro de outro loop para percorrer todas as posições

Array bidimensional

```
1  #include <stdio.h>
2
3  #define tamanho 3
4  int main() {
5
6      //declarando um array bidimensional
7      float matriz[tamanho][tamanho] = {{0.0, 0.1, 0.2}, {1.0, 1.1, 1.2}, {2.0, 2.1, 2.2}};
8
9      int i,j; // 1 variável para cada dimensão
10     for(i=0;i<tamanho;i++) {
11         for(j=0;j<tamanho;j++) {
12             printf("%.1f - ", matriz[i][j]);
13         }
14         printf("\n");
15     }
16
17     return 0;
18 }
```

Array bidimensional

```
danilo@danilo-PC:~/Desktop$ ./matriz1
0.0 - 0.1 - 0.2 -
1.0 - 1.1 - 1.2 -
2.0 - 2.1 - 2.2 -
```

```
1  #include <stdio.h>
2
3  #define tamanho 3
4  int main() {
5
6      //declarando um array bidimensional
7      float matriz[tamanho][tamanho] = {{0.0, 0.1, 0.2}, {1.0, 1.1, 1.2}, {2.0, 2.1, 2.2}};
8
9      int i,j; // 1 variável para cada dimensão
10     for(i=0;i<tamanho;i++) {
11         for(j=0;j<tamanho;j++) {
12             printf("%.1f - ", matriz[i][j]);
13         }
14         printf("\n");
15     }
16
17     return 0;
18 }
```

Array tridimensional

- Cubo com 3 dimensões
- Como é feita a declaração?

Array tridimensional

- Cubo com 3 dimensões
- Como é feita a declaração?
 - Teremos `[][][]`
 - 1 `[]` para cada dimensão

Array tridimensional

- Cubo com 3 dimensões
- Como percorremos todas as dimensões?

Array tridimensional

- Cubo com 3 dimensões
- Como percorremos todas as dimensões?
 - Precisamos de 3 loops, 1 aninhado/dentro do outro

Array

- Como deve ser o código?
 - Para gravar em cada posição de um array bidimensional 3x3 o resultado da multiplicação dos índices
 - Imprimir o valor de todas as posições do array

Array

```
1  #include <stdio.h>
2
3  #define tamanho 3
4  int main() {
5
6      //declarando um array bidimensional
7      int matriz[tamanho][tamanho];
8
9      int i,j; // 1 variável para cada dimensão
10     for(i=0;i<tamanho;i++) {
11         for(j=0;j<tamanho;j++) {
12             matriz[i][j] = i * j;
13         }
14     }
15
16     for(i=0;i<tamanho;i++) {
17         for(j=0;j<tamanho;j++) {
18             printf("%d - ", matriz[i][j]);
19         }
20         printf("\n");
21     }
22
23     return 0;
24 }
```

```
1  #include <stdio.h>
2
3  #define tamanho 3
4  int main() {
5
6      //declarando um array bidimensional
7      int matriz[tamanho][tamanho];
8
9      int i,j; // 1 variável para cada dimensão
10     for(i=0;i<tamanho;i++) {
11         for(j=0;j<tamanho;j++) {
12             matriz[i][j] = i * j;
13         }
14     }
15
16     for(i=0;i<tamanho;i++) {
17         for(j=0;j<tamanho;j++) {
18             printf("%d - ", matriz[i][j]);
19         }
20         printf("\n");
21     }
22
23     return 0;
24 }
```

danilo@danilo-PC:~/Desktop\$./matriz2

0 - 0 - 0 -

0 - 1 - 2 -

0 - 2 - 4 -

Array

- Como deve ser o código?
 - Para criar um array bidimensional com 2 linhas e 5 colunas
 - Pedir para a pessoa digitar 5 números e guardar na primeira linha
 - Colocar na segunda linha o dobro do número digitado
 - Imprimir o valor de todas as posições do array

Array

```
1  #include <stdio.h>
2
3  #define linha 2
4  #define coluna 5
5  int main() {
6
7      //declarando um array bidimensional
8      int matriz[linha][coluna];
9
10     int i,j; // 1 variável para cada dimensão
11
12     printf("Vou pedir para digitar 5 números\n");
13
14     for(i=0;i<coluna;i++) {
15         printf("Digite o numero %d: ",i+1);
16         scanf("%d", &matriz[0][i]);
17
18         matriz[1][i] = 2 * matriz[0][i];
19     }
20
21     for(i=0;i<linha;i++) {
22         for(j=0;j<coluna;j++) {
23             printf("%d - ", matriz[i][j]);
24         }
25         printf("\n");
26     }
27
28     return 0;
29 }
```

```

1 #include <stdio.h>
2
3 #define linha 2
4 #define coluna 5
5 int main() {
6
7     //declarando um array bidimensional
8     int matriz[linha][coluna];
9
10    int i,j; // 1 variável para cada dimensão
11
12    printf("Vou pedir para digitar 5 números\n");
13
14    for(i=0;i<coluna;i++) {
15        printf("Digite o numero %d: ",i+1);
16        scanf("%d", &matriz[0][i]);
17
18        matriz[1][i] = 2 * matriz[0][i];
19    }
20
21    for(i=0;i<linha;i++) {
22        for(j=0;j<coluna;j++) {
23            printf("%d - ", matriz[i][j]);
24        }
25        printf("\n");
26    }
27
28    return 0;
29 }

```

```

dani@danilo-PC:~/Desktop$ ./matriz3
Vou pedir para digitar 5 números
Digite o numero 1: 1
Digite o numero 2: 2
Digite o numero 3: 3
Digite o numero 4: 4
Digite o numero 5: 5
1 - 2 - 3 - 4 - 5 -
2 - 4 - 6 - 8 - 10 -

```

Exercício 4

- Fazer uma mini batalha naval
- Menu com 2 opções
 - 1) Jogar
 - Fazer um código que possui uma matriz 5x5, com todas as posições com o valor _ (underline)
 - Antes de iniciar o jogo, escolher aleatoriamente 5 posições diferentes da matriz onde estarão posicionados cada um dos barcos de tamanho 1
 - Imprimir a matriz do tabuleiro
 - Ficar em loop pedindo para a pessoa digitar o valor de linha e coluna até acertar todos os barcos
 - Se acertar o barco, substituir por "x" e informar que acertou
 - Se não acertar o barco, substituir por "o" e informar que errou
 - Se atirar numa posição repetida, informar a pessoa
 - Ao finalizar informar quantas posições do tabuleiro não foram atingidas
 - Voltar ao menu
 - 2) Sair
 - Finalizar o código
- Entrega até: 05/09

Perguntas?

Obrigado!

Até a próxima aula!