

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

Danilo Rios
26/09/2025

Conteúdo prévio

- Biblioteca
- Função Main
- printf
- Variáveis
- Constantes
- scanf
- Condicionais
 - if, else, ternário, switch
- Loop
 - for, while, do while

Array

- O que é?

Array

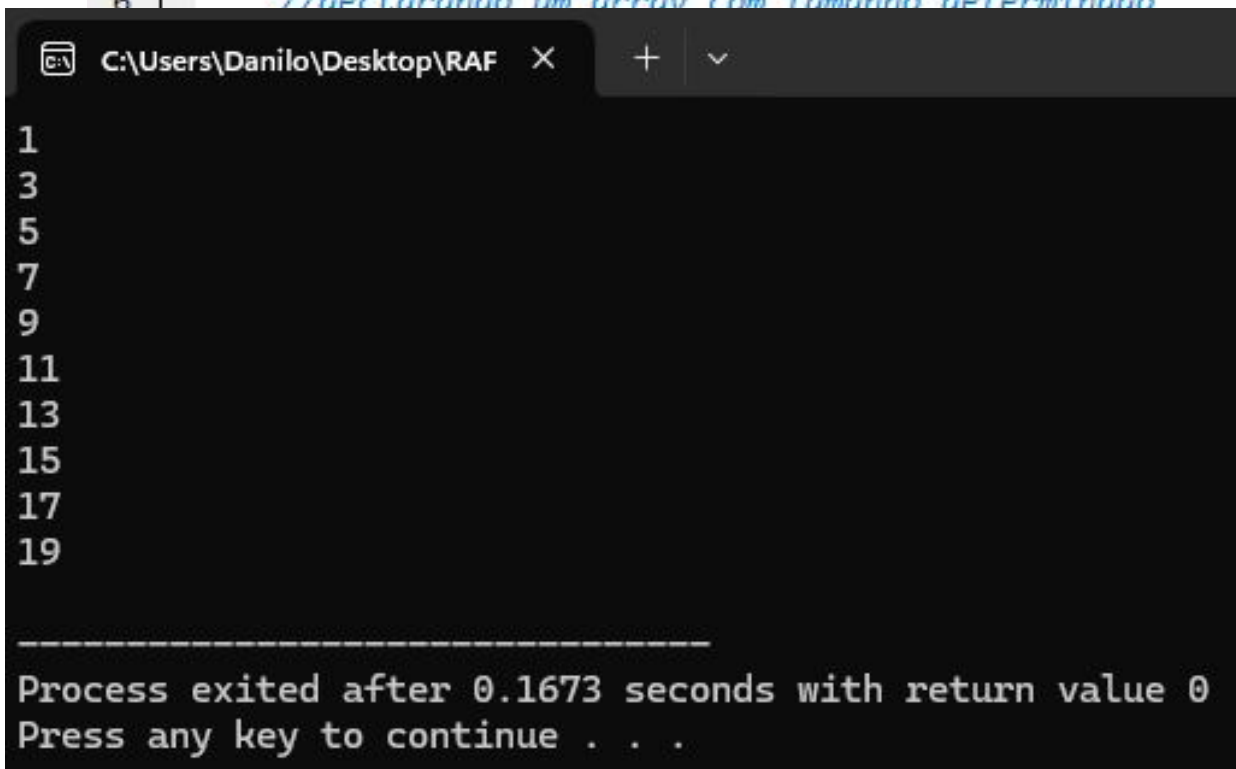
- O que é?
 - Vetor
 - É uma variável que possui vários valores

Array

```
1  #include <stdio.h>
2
3  #define tamanho 10
4  int main() {
5
6      //declarando um array com tamanho determinado
7      int vetor[tamanho];
8
9      vetor[0] = 1;
10     vetor[1] = 3;
11     vetor[2] = 5;
12     vetor[3] = 7;
13     vetor[4] = 9;
14     vetor[5] = 11;
15     vetor[6] = 13;
16     vetor[7] = 15;
17     vetor[8] = 17;
18     vetor[9] = 19;
19
20     int i;
21     for(i=0;i<tamanho;i++) {
22         printf("%d\n", vetor[i]);
23     }
24
25     return 0;
26 }
```

Array

```
1 #include <stdio.h>
2
3 #define tamanho 10
4 int main() {
5
6     //declarando um array com tamanho determinado
```



```
C:\Users\Danilo\Desktop\RAF X + v
1
3
5
7
9
11
13
15
17
19
-----
Process exited after 0.1673 seconds with return value 0
Press any key to continue . . .
```

```
25     return 0;
26 }
```

```
1  #include <stdio.h>
2
3  #define tamanho 10
4  int main() {
5
6      //declarando um array sem determinar o tamanho e atribuindo valores
7      //tamanho do array será determinado pela quantidade de valores atribuídos
8      int vetor[] = {0,1,2,3,4,5,6,7,8,9};
9
10     //declarando um array com tamanho determinado (10) e atribuindo valores
11     //atribuindo uma quantidade menor de valores do que o tamanho do array
12     int vetor2[tamanho] = {0,1,2,3,4,5};
13
14     int i;
15     for(i=0;i<tamanho;i++) {
16         printf("%d - %d\n", vetor[i], vetor2[i]);
17     }
18
19     return 0;
20 }
```

```
1  #include <stdio.h>
2
3  #define tam 10
4  int main()
5
6  //dec
7  //tam
8  int v[10]
9
10 //dec
11 //atr
12 int v[10]
13
14 int i
15 for(i=0; i<10; i++)
16     p[i] = i;
17
18
19 return 0;
20 }
```

C:\Users\Danilo\Desktop\RAF

0 - 0

1 - 1

2 - 2

3 - 3

4 - 4

5 - 5

6 - 0

7 - 0

8 - 0

9 - 0

Process exited after 0.16 seconds with return value 0
Press any key to continue . . .

o valores
res atribuidos

uindo valores
nho do array

Array

```
1  #include <stdio.h>
2
3  #define tamanho 5
4  int main() {
5
6      int vetor[tamanho];
7
8      printf("Vou pedir para digitar 5 numeros ...\n");
9
10     printf("Digite o numero 1: ");
11     scanf("%d", &vetor[0]);
12     printf("Digite o numero 2: ");
13     scanf("%d", &vetor[1]);
14     printf("Digite o numero 3: ");
15     scanf("%d", &vetor[2]);
16     printf("Digite o numero 4: ");
17     scanf("%d", &vetor[3]);
18     printf("Digite o numero 5: ");
19     scanf("%d", &vetor[4]);
20
21     printf("\n\nNumeros digitados: ");
22     int i;
23     for(i=0;i<tamanho;i++) {
24         printf("%d", vetor[i]);
25
26         if(i != tamanho - 1) {
27             printf(", ");
28         }
29     }
30
31     return 0;
32 }
```

```

1 #include <stdio.h>
2
3 #define tamanho 5
4 int main() {
5
6     int vetor[tamanho];
7
8     printf("Vou pedir para digitar 5 numeros ...\n");
9
10    printf("Digite o numero 1: ");
11    scanf("%d", &vetor[0]);
12    printf("Digite o numero 2: ");
13    scanf("%d", &vetor[1]);
14    printf("Digite o numero 3: ");
15    scanf("%d", &vetor[2]);
16    printf("Digite o numero 4: ");
17    scanf("%d", &vetor[3]);
18    printf("Digite o numero 5: ");
19    scanf("%d", &vetor[4]);
20
21    printf("\n\nNumeros digitados: ");
22    int i;
23    for(i=0;i<tamanho;i++) {
24        printf("%d", vetor[i]);
25
26        if(i != tamanho - 1) {
27            printf(", ");
28        }
29    }
30
31    return 0;
32 }

```

```

danilo@danilo-PC:~/Desktop$ ./PreencherArray

```

```

Vou pedir para digitar 5 numeros ...

```

```

Digite o numero 1: 30

```

```

Digite o numero 2: 19

```

```

Digite o numero 3: 77

```

```

Digite o numero 4: 95

```

```

Digite o numero 5: 1

```

```

Numeros digitados: 30, 19, 77, 95, 1danilo@danilo-PC:~/Desktop$ 

```

Array

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define aleatorio 100
5  #define digitado 5
6
7  int main() {
8
9      int vetorAleatorio[aleatorio];
10     int vetorDigitado[digitado];
11     int i;
12
13     for(i=0;i<aleatorio;i++) {
14         vetorAleatorio[i] = rand();
15     }
16
17     printf("Vou pedir para digitar 5 numeros ...\n");
18
19     for(i=0;i<digitado;i++) {
20         printf("Digite o numero %d: ", i+1);
21         scanf("%d", &vetorDigitado[i]);
22     }
23
24     printf("\n\nNumeros sorteados:\n");
25
26     for(i=0;i<digitado;i++) {
27         printf("%d\n", vetorAleatorio[vetorDigitado[i]%aleatorio]);
28     }
29
30     return 0;
31 }
```

```

2  #include <stdlib.h>
3
4  #define aleatorio 100
5  #define digitado 5
6
7  int main() {
8
9      int vetorAleatorio[aleatorio];
10     int vetorDigitado[digitado];
11     int i;
12
13     for(i=0;i<aleatorio;i++) {
14         vetorAleatorio[i] = rand();
15     }
16
17     printf("Vou pedir para digitar 5 numeros ...\n");
18
19     for(i=0;i<digitado;i++) {
20         printf("Digite o numero %d: ", i+1);
21         scanf("%d", &vetorDigitado[i]);
22     }
23
24     printf("\n\nNumeros sorteados:\n");
25
26     for(i=0;i<digitado;i++) {
27         printf("%d\n", vetorAleatorio[vetorDigitado[i]%aleatorio]);
28     }
29
30     return 0;
31 }

```

```

danilo@danilo-PC:~/Desktop$ ./Aleatorio
Vou pedir para digitar 5 numeros ...
Digite o numero 1: 1000
Digite o numero 2: 316
Digite o numero 3: 44
Digite o numero 4: 293
Digite o numero 5: 12

Numeros sorteados:
1804289383
846930886
1957747793
1714636915
1681692777

```

Array

```
1  #include <stdio.h>
2
3  #define tamanho 10
4
5  int main() {
6
7      int vetorSorteados[tamanho];
8      int vetorApostados[tamanho];
9      int vetorAcertos[tamanho];
10     int i,j;
11     int qtdAcertos = 0;
12
13     printf("Vou pedir para digitar os %d numeros sorteados ...\n", tamanho);
14
15     for(i=0;i<tamanho;i++) {
16         printf("Digite o numero %d: ", i+1);
17         scanf("%d", &vetorSorteados[i]);
18     }
19
20     printf("\nVou pedir para digitar os %d numeros apostados ...\n", tamanho);
21
22     for(i=0;i<tamanho;i++) {
23         printf("Digite o numero %d: ", i+1);
24         scanf("%d", &vetorApostados[i]);
25     }
26 }
```

Array

```
27     for(i=0;i<tamanho;i++) {
28         for(j=0;j<tamanho;j++) {
29             if(vetorSorteados[i] == vetorApostados[j]) {
30                 vetorAcertos[qtdAcertos] = vetorSorteados[i];
31                 qtdAcertos++;
32             }
33         }
34     }
35
36     printf("\nResultado:\nQuantidade de numeros iguais:%d", qtdAcertos);
37     if(qtdAcertos > 0) {
38         printf("\nNumeros: ");
39         for(i=0;i<qtdAcertos;i++) {
40             printf("%d", vetorAcertos[i]);
41
42             if(i < qtdAcertos-1) {
43                 printf(", ");
44             }
45         }
46     }
47
48     printf("\n");
49
50     return 0;
51 }
```



```

1 #include <stdio.h>
2
3 #define tamanho 10
4
5 int main() {
6
7     int vetorSorteados[tamanho];
8     int vetorApostados[tamanho];
9     int vetorAcertos[tamanho];
10    int i,j;
11    int qtdAcertos = 0;
12
13    printf("Vou pedir para digitar os %d numeros sorteados ...\n", tamanho);
14
15    for(i=0;i<tamanho;i++) {
16        printf("Digite o numero %d: ", i+1);
17        scanf("%d", &vetorSorteados[i]);
18    }
19
20    printf("\nVou pedir para digitar os %d numeros apostados ...\n", tamanho);
21
22    for(i=0;i<tamanho;i++) {
23        printf("Digite o numero %d: ", i+1);
24        scanf("%d", &vetorApostados[i]);
25    }
26
27    for(i=0;i<tamanho;i++) {
28        for(j=0;j<tamanho;j++) {
29            if(vetorSorteados[i] == vetorApostados[j]) {
30                vetorAcertos[qtdAcertos] = vetorSorteados[i];
31                qtdAcertos++;
32            }
33        }
34    }
35
36    printf("\nResultado:\nQuantidade de numeros iguais:%d", qtdAcertos);
37    if(qtdAcertos > 0) {
38        printf("\nNumeros: ");
39        for(i=0;i<qtdAcertos;i++) {
40            printf("%d", vetorAcertos[i]);
41
42            if(i < qtdAcertos-1) {
43                printf(", ");
44            }
45        }
46    }
47
48    printf("\n");
49
50    return 0;
51 }

```

```

daniilo@daniilo-PC:~/Desktop$ ./Aposta
Vou pedir para digitar os 10 numeros sorteados ...
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
Digite o numero 6: 6
Digite o numero 7: 7
Digite o numero 8: 8
Digite o numero 9: 9
Digite o numero 10: 10

```

```

Vou pedir para digitar os 10 numeros apostados ...
Digite o numero 1: 2
Digite o numero 2: 4
Digite o numero 3: 6
Digite o numero 4: 8
Digite o numero 5: 10
Digite o numero 6: 12
Digite o numero 7: 14
Digite o numero 8: 16
Digite o numero 9: 18
Digite o numero 10: 20

```

```

Resultado:
Quantidade de numeros iguais:5
Numeros: 2, 4, 6, 8, 10

```

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 bidim

```
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 bidime

```
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 -

```

Ponteiro

- Armazena o endereço de memória em que está alguma variável

Ponteiro

- Identificado pelo *
- Asterisco
 - Não é “asteristico”

Ponteiro

```
1  #include <stdio.h>
2
3  int main() {
4
5      int numero = 5;
6      int *ponteiro = &numero;
7
8      printf("numero:\n");
9      printf("Valor: %d\n", numero);
10     printf("Endereco: %p\n", &numero);
11
12     printf("\n\n");
13     printf("ponteiro:\n");
14     printf("Valor: %p\n", ponteiro);
15     printf("Valor apontado: %d\n", *ponteiro);
16     printf("Endereco: %p\n", &ponteiro);
17
18     return 0;
19 }
```


Ponteiro

```
1  #include <stdio.h>
2
3  int main() {
4
5      int numero = 5;
6      int *ponteiro = &numero;
7
8      printf("numero:\n");
9      printf("Valor: %d\n", numero);
10     printf("Endereco: %p\n", &numero);
11
12     printf("\n\n");
13     printf("ponteiro:\n");
14     printf("Valor: %p\n", ponteiro);
15     printf("Valor apontado: %d\n", *ponteiro);
16     printf("Endereco: %p\n", &ponteiro);
17
18     return 0;
19 }
```

```
numero:
Valor: 5
Endereco: 000000000062FE4C
```

```
ponteiro:
Valor: 000000000062FE4C
Valor apontado: 5
Endereco: 000000000062FE40
```

```
-----
Process exited after 0.1816 seconds with return value 0
Press any key to continue . . .
```

Ponteiro

- Tipo do ponteiro
 - O tipo do ponteiro tem que ser o mesmo da variável

Ponteiro

```
1  #include <stdio.h>
2
3  int main() {
4
5      int numero = 5;
6      int *ponteiro = &numero;
7
8      printf("Valor do numero: %d\n", numero);
9
10     numero++;
11     printf("Valor do numero: %d\n", numero);
12
13     //colocando 100 no endereço de memória apontado
14     *ponteiro = 100;
15     printf("Valor do numero: %d\n", numero);
16
17     return 0;
18 }
```

Ponteiro

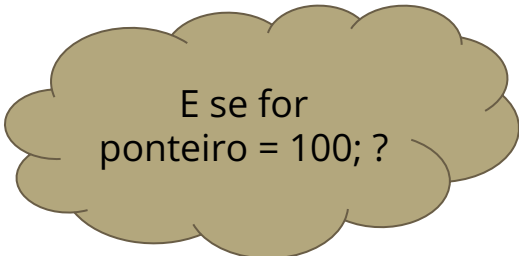
```
1  #include <stdio.h>
2
3  int main() {
4
5      int numero = 5;
6      int *ponteiro = &numero;
7
8      printf("Valor do numero: %d\n", numero);
9
10     numero++;
11     printf("Valor do numero: %d\n", numero);
12
13     //colocando 100 no endereço de memória apontado
14     *ponteiro = 100;
15     printf("Valor do numero: %d\n", numero);
16
17     return 0;
18 }
```

```
Valor do numero: 5
Valor do numero: 6
Valor do numero: 100
```

```
-----
Process exited after 0.1703 seconds with return value 0
Press any key to continue . . .
```

Ponteiro

```
1  #include <stdio.h>
2
3  int main() {
4
5      int numero = 5;
6      int *ponteiro = &numero;
7
8      printf("Valor do numero: %d\n", numero);
9
10     numero++;
11     printf("Valor do numero: %d\n", numero);
12
13     //colocando 100 no endereço de memória apontado
14     *ponteiro = 100;
15     printf("Valor do numero: %d\n", numero);
16
17     return 0;
18 }
```



E se for
ponteiro = 100; ?

Ponteiro

```
1  #include <stdio.h>
2
3  int main() {
4
5      int numero = 5;
6      int *ponteiro = &numero;
7
8      printf("Ponteiro esta apontando para %p\n", ponteiro);
9      printf("Valor apontado: %d\n", *ponteiro);
10
11     printf("Valor do numero: %d\n", numero);
12
13     numero++;
14     printf("Valor do numero: %d\n", numero);
15
16     //colocando 100 no endereço de memória apontado
17     ponteiro = 100;
18     printf("Valor do numero: %d\n", numero);
19
20     printf("Ponteiro esta apontando para %p\n", ponteiro);
21     printf("Valor apontado: %d\n", *ponteiro);
22
23     return 0;
24 }
```

Ponteiro

```
1 #include <stdio.h>
2
3 int main() {
4
5     int numero = 5;
6     int *ponteiro = &numero;
7
8     printf("Ponteiro esta apontando para %p\n", ponteiro);
9     printf("Valor apontado: %d\n", *ponteiro);
10
11     printf("Valor do numero: %d\n", numero);
12
13     numero++;
14     printf("Valor do numero: %d\n", numero);
15
16     //colocando 100 no endereço de memória apontado
17     ponteiro = 100;
18     printf("Valor do numero: %d\n", numero);
19
20     printf("Ponteiro esta apontando para %p\n", ponteiro);
21     printf("Valor apontado: %d\n", *ponteiro);
22
23     return 0;
24 }
```

```
Ponteiro esta apontando para 000000000062FE44
Valor apontado: 5
Valor do numero: 5
Valor do numero: 6
Valor do numero: 6
Ponteiro esta apontando para 0000000000000064
```

```
-----
Process exited after 0.9711 seconds with return value 3221225477
Press any key to continue . . .
```

Ponteiro + Array

```
1  #include <stdio.h>
2
3  int main() {
4
5      int i;
6      int numeros[] = {1,2,3,4,5,6,7,8,9,10};
7      int *ponteiro = &numeros; //são varios valores, aponta para qual?
8
9      printf("Ponteiro esta apontando para %p\n", ponteiro);
10
11     for(i=0;i<10;i++) {
12         //imprimindo os endereços de memória de cada posição
13         printf("%d = %p\n", numeros[i], &numeros[i]);
14     }
15
16     return 0;
17 }
```


Ponteiro + Array

```
1  #include <stdio.h>
2
3  int main() {
4
5      int i;
6      int numeros[] = {1,2,3,4,5,6,7,8,9,10};
7      int *ponteiro = &numeros; //são varios valores, aponta para qual?
8
9      printf("Ponteiro esta apontando para %p\n", ponteiro);
10
11     for(i=0;i<10;i++) {
12         //imprimindo os endereços de memória de cada posição
13         printf("%d = %p\n", numeros[i], &numeros[i]);
14     }
15
16     return 0;
17 }
```

```
Ponteiro esta apontando para 000000000062FE10
1 = 000000000062FE10
2 = 000000000062FE14
3 = 000000000062FE18
4 = 000000000062FE1C
5 = 000000000062FE20
6 = 000000000062FE24
7 = 000000000062FE28
8 = 000000000062FE2C
9 = 000000000062FE30
10 = 000000000062FE34
```

```
-----
Process exited after 0.4777 seconds with return value 0
Press any key to continue . . .
```

Ponteiro + Array

```
Ponteiro esta apontando para 000000000062FE10
1 = 000000000062FE10
2 = 000000000062FE14
3 = 000000000062FE18
4 = 000000000062FE1C
5 = 000000000062FE20
6 = 000000000062FE24
7 = 000000000062FE28
8 = 000000000062FE2C
9 = 000000000062FE30
10 = 000000000062FE34
```

```
-----
Process exited after 0.4777 seconds with return value 0
Press any key to continue . . .
```

- Existe uma sequência
- Pula de 4 em 4 bytes
- Será que consigo mexer no ponteiro e acessar cada posição?

Ponteiro + Array

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

Ponteiro + Array

```
numeros[0] = 1  
numeros[1] = 2  
numeros[2] = 3  
numeros[3] = 4  
numeros[4] = 5
```

Process exited after 0.1765 seconds with return value 0
Press any key to continue . . .

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

Ponteiro

```
1  #include <stdio.h>
2
3  #define tamanho 10
4
5  int main() {
6      int i;
7      int numeros[tamanho];
8      int *ponteiro = numeros;
9
10     printf("Vou pedir para digitar 10 numeros\n");
11
12     for(i=0;i<tamanho;i++) {
13         printf("Digite #%d numero: ", i+1);
14         scanf("%d", &*(ponteiro + i));
15     }
16
17     for(i=0;i<tamanho;i++) {
18         printf("%d\n", numeros[i]);
19     }
20
21     return 0;
22 }
```

Struct

- O que é?

Struct

- O que é?
 - Estrutura
 - E o que significa?

Struct

- É uma variável composta por variáveis

Struct

```
1  #include <stdio.h>
2
3  int main() {
4
5      struct Data {
6          int dia;
7          int mes;
8          int ano;
9      };
10
11      struct Data dataNascimento = {1, 1, 2000};
12
13      struct Data natal;
14      natal.dia = 25;
15      natal.mes = 12;
16      natal.ano = 2000;
17
18      printf("Data de nascimento: %d/%d/%d\n", dataNascimento.dia, dataNascimento.mes, dataNascimento.ano);
19      printf("Natal: %d/%d/%d\n", natal.dia, natal.mes, natal.ano);
20
21      return 0;
22  }
```

Struct

```
1  #include <stdio.h>
2
3  int main() {
4
5      struct Data {
6          int dia;
7          int mes;
8          int ano;
9      };
10
11      struct Data dataNascimento = {1, 1, 2000};
12
13      struct Data natal;
14      natal.dia = 25;
15      natal.mes = 12;
16      natal.ano = 2000;
17
18      printf("Data de nascimento: %d/%d/%d\n", dataNascimento.dia, dataNascimento.mes, dataNascimento.ano);
19      printf("Natal: %d/%d/%d\n", natal.dia, natal.mes, natal.ano);
20
21      return 0;
22 }
```

```
danilo@danilo-PC:~/Desktop$ ./struct1
Data de nascimento: 1/1/2000
Natal: 25/12/2000
```

Struct

- E o ponteiro de uma struct?

```
1  #include <stdio.h>
2
3  int main() {
4
5      struct Data {
6          int dia;
7          int mes;
8          int ano;
9      };
10
11     struct Data dataNascimento = {1, 1, 2000};
12
13     struct Data *ponteiro = &dataNascimento;
14
15     printf("Endereco dataNascimento = %p\n", &dataNascimento);
16     printf("Endereco apontado pelo ponteiro = %p\n", ponteiro);
17
18     printf("Data de nascimento: %d/%d/%d\n", ponteiro->dia, ponteiro->mes, ponteiro->ano);
19
20     return 0;
21 }
```

```
1  #include <stdio.h>
```

```
2  
3  int main() {
```

```
4  
5      struct Data {
```

```
6          int dia;
```

```
7          int mes;
```

```
8          int ano;
```

```
9      };
```

```
10  
11     struct Data dataNascimento = {1, 1, 2000};
```

```
12  
13     struct Data *ponteiro = &dataNascimento;
```

```
14  
15     printf("Endereco dataNascimento = %p\n", &dataNascimento);
```

```
16     printf("Endereco apontado pelo ponteiro = %p\n", ponteiro);
```

```
17  
18     printf("Data de nascimento: %d/%d/%d\n", ponteiro->dia, ponteiro->mes, ponteiro->ano);
```

```
19  
20     return 0;
```

```
21 }
```

```
danilo@danilo-PC:~/Desktop$ ./struct2
```

```
Endereco dataNascimento = 0x7fff9d8c91dc
```

```
Endereco apontado pelo ponteiro = 0x7fff9d8c91dc
```

```
Data de nascimento: 1/1/2000
```

Stru

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5
6      struct Aluno {
7          char nome[100];
8          float media;
9      };
10
11     struct Aluno aluno;
12
13     printf("Digite o nome do aluno: ");
14     fgets(aluno.nome, sizeof(aluno.nome), stdin);
15
16     printf("Digite a media do aluno: ");
17     scanf("%f", &aluno.media);
18
19     printf("\nAluno: %sMedia: %.2f\n", aluno.nome, aluno.media);
20
21     return 0;
22 }
```

Struct

- Como deve ser o código?
 - Pedir para a pessoa digitar o nome da pessoa, o dia de nascimento, o mês de nascimento e o ano de nascimento e armazenar isso em uma struct
 - Depois imprimir:
 - Nome: <nome>
 - Data de nascimento: <dd>/<mm>/<aaaa>

```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main() {
5
6      struct Pessoa {
7          char nome[100];
8          int dia;
9          int mes;
10         int ano;
11     };
12
13     struct Pessoa pessoa;
14
15     printf("Digite o nome da pessoa: ");
16     fgets(pessoa.nome, sizeof(pessoa.nome), stdin);
17     pessoa.nome[strcspn(pessoa.nome, "\n")] = '\0';
18
19     printf("Digite o dia de nascimento: ");
20     scanf("%d", &pessoa.dia);
21
22     printf("Digite o mes de nascimento: ");
23     scanf("%d", &pessoa.mes);
24
25     printf("Digite o ano de nascimento: ");
26     scanf("%d", &pessoa.ano);
27
28     printf("\nNome: %s\nData de nascimento: %d/%d/%d\n", pessoa.nome, pessoa.dia, pessoa.mes, pessoa.ano);
29
30     return 0;
31 }
```



```
1 #include <stdio.h>
2 #include <string.h>
```

```
3
4 int main() {
```

```
5
6     struct Pessoa {
7         char nome[100];
8         int dia;
9         int mes;
10        int ano;
11    };
```

```
12
13    struct Pessoa pessoa;
```

```
14
15    printf("Digite o nome da pessoa: ");
16    fgets(pessoa.nome, sizeof(pessoa.nome), stdin);
17    pessoa.nome[strcspn(pessoa.nome, "\n")] = '\0';
```

```
18
19    printf("Digite o dia de nascimento: ");
20    scanf("%d", &pessoa.dia);
```

```
21
22    printf("Digite o mes de nascimento: ");
23    scanf("%d", &pessoa.mes);
```

```
24
25    printf("Digite o ano de nascimento: ");
26    scanf("%d", &pessoa.ano);
```

```
27
28    printf("\nNome: %s\nData de nascimento: %d/%d/%d\n", pessoa.nome, pessoa.dia, pessoa.mes, pessoa.ano);
```

```
29
30    return 0;
```

```
31 }
```

```
danilo@danilo-PC:~/Desktop$ ./struct4
```

Digite o nome da pessoa: Pessoa

Digite o dia de nascimento: 01

Digite o mes de nascimento: 01

Digite o ano de nascimento: 2000

Nome: Pessoa

Data de nascimento: 1/1/2000

Struc

```
1  #include <stdio.h>
2
3  #define tamanho 10
4
5  int main() {
6
7      struct Numero {
8          int numero;
9          char p0uI;
10     };
11
12     struct Numero numeros[tamanho];
13     int i;
14
15     for(i=0;i<tamanho;i++) {
16         printf("Digite um numero: ");
17         scanf("%d", &numeros[i].numero);
18
19         numeros[i].p0uI = (numeros[i].numero % 2 == 0 ? 'P' : 'I');
20     }
21
22     printf("\n");
23
24     for(i=0;i<tamanho;i++) {
25         printf("Numero %d eh %c\n", numeros[i].numero, numeros[i].p0uI);
26     }
27
28     return 0;
29 }
```

Struc

```
1  #include <stdio.h>
2
3  #define tamar
4
5  int main() {
6      struct Nu
7          int r
8          char
9      };
10
11      struct Nu
12      int i;
13      for(i=0;i
14          print
15          scanf
16          numero
17      }
18      printf("\
19      for(i=0;i
20          print
21      }
22      return 0;
23
24 }
```

danilo@danilo-PC:~/Desktop\$./struct5

Digite um numero: 10
Digite um numero: 9
Digite um numero: 8
Digite um numero: 7
Digite um numero: 6
Digite um numero: 5
Digite um numero: 4
Digite um numero: 3
Digite um numero: 2
Digite um numero: 1

Numero 10 eh P
Numero 9 eh I
Numero 8 eh P
Numero 7 eh I
Numero 6 eh P
Numero 5 eh I
Numero 4 eh P
Numero 3 eh I
Numero 2 eh P
Numero 1 eh I

'P' : 'I');

numeros[i].p0uI);

Perguntas?

Obrigado!

Até a próxima aula!