
Raciocínio Algorítmico e Fundamentos da Computação

— Danilo Rios —
16/05/2025

Array

- O que é?

Array

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

Array

- Como assim vários valores?
 - Se atribuir um segundo valor para a variável substitui o anterior
- Como isso é possível?

Array

- É uma variável que possui uma quantidade definida de posições
- Cada valor é salvo em uma posição diferente
- Exemplo: Array de números ímpares

1	3	5	7	9	11	13	15	...	99
---	---	---	---	---	----	----	----	-----	----

Posição: 0 1 2 3 4 5 6 7 ... n

Atenção!

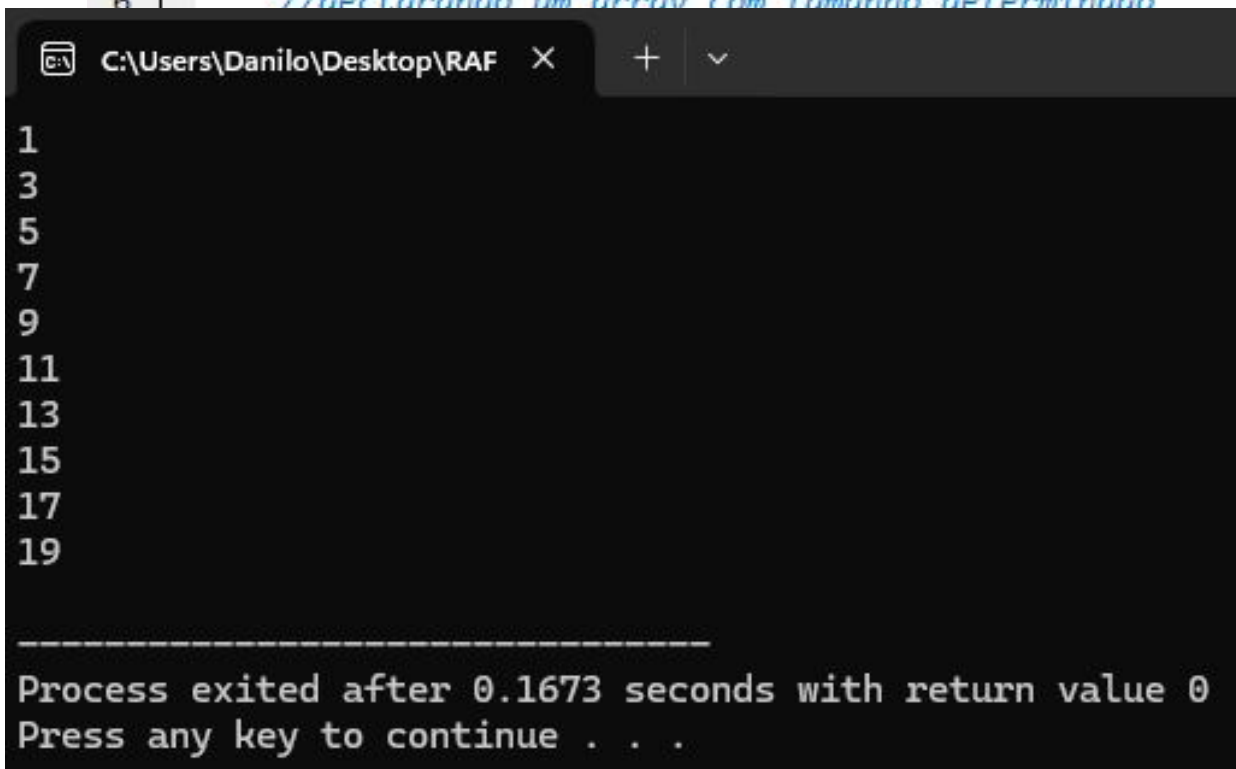
- A primeira posição do array é a posição ZERO!
 - Não é a posição 1

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     {
17         p
18     }
19
20     return 0;
}
```

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

- 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 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

```
1  #include <stdio.h>
2
3  #define tamanho 3
4  int main() {
5
6      //declarando um array bidimensional
7      //todas as dimensões devem ser declaradas OU pelo menos a última
8      int matriz[tamanho][tamanho] = {{0,1,2},{3,4,5},{6,7,8}};
9
10     int i,j;
11     for(i=0;i<tamanho;i++) {
12         for(j=0;j<tamanho;j++) {
13             printf("%d - ", matriz[i][j]);
14         }
15         printf("\n");
16     }
17
18     return 0;
19 }
```

```
1  #include <stdio.h>
2
3  #define tamanho 3
4  int main() {
5
6      //declarando um array bidimensional
7      //todas as dimensões devem ser declaradas OU pelo menos a última
8      int array[tamanho][tamanho] = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8}};
9
10     for (int i = 0; i < tamanho; i++) {
11         for (int j = 0; j < tamanho; j++) {
12             printf("%d - ", array[i][j]);
13             if (j == tamanho - 1) {
14                 printf("\n");
15             }
16         }
17     }
18     return 0;
19 }
```

C:\Users\Danilo\Desktop\RAF X + v

```
0 - 1 - 2 -
3 - 4 - 5 -
6 - 7 - 8 -

-----
Process exited after 0.4624 seconds with return value 0
Press any key to continue . . . |
```

Array tridimensional

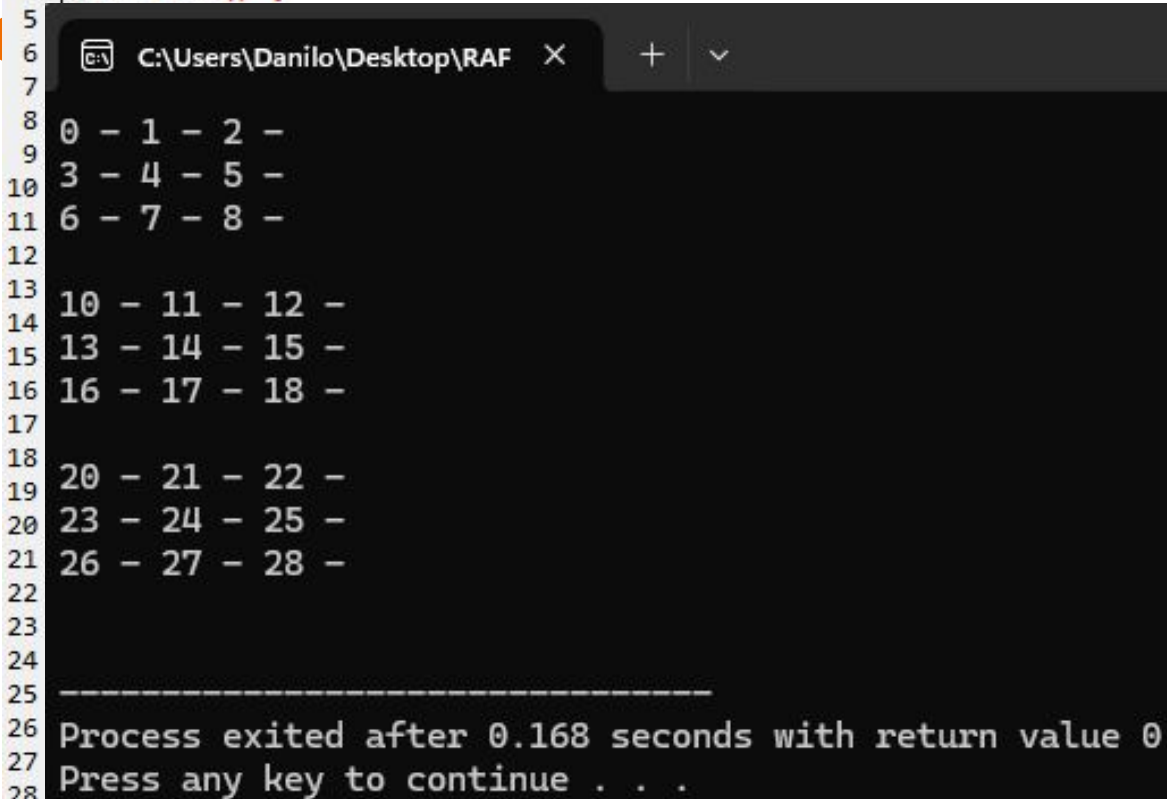
- Como que fazemos um array tridimensional?
- Como percorremos um array tridimensional?

Array tric

```
1  #include <stdio.h>
2
3  #define tamanho 3
4  int main() {
5
6      //declarando um array bidimensional
7      //todas as dimensões devem ser declaradas OU pelo menos as 2 últimas
8      int cubo[tamanho][tamanho][tamanho] = {
9          {
10             {0,1,2},{3,4,5},{6,7,8}
11         },
12         {
13             {10,11,12},{13,14,15},{16,17,18}
14         },
15         {
16             {20,21,22},{23,24,25},{26,27,28}
17         }
18     };
19
20     int i,j,k;
21     for(i=0;i<tamanho;i++) {
22         for(j=0;j<tamanho;j++) {
23             for(k=0;k<tamanho;k++) {
24                 printf("%d - ", cubo[i][j][k]);
25             }
26             printf("\n");
27         }
28         printf("\n");
29     }
30
31     return 0;
32 }
```


Array trick

```
1 #include <stdio.h>
2
3 #define tamanho 3
4 int main() {
```



```
5
6 0 - 1 - 2 -
7
8 3 - 4 - 5 -
9
10 6 - 7 - 8 -
11
12
13 10 - 11 - 12 -
14
15 13 - 14 - 15 -
16
17 16 - 17 - 18 -
18
19
20 20 - 21 - 22 -
21
22 23 - 24 - 25 -
23
24 26 - 27 - 28 -
25
26 -----
27 Process exited after 0.168 seconds with return value 0
28 Press any key to continue . . .
```

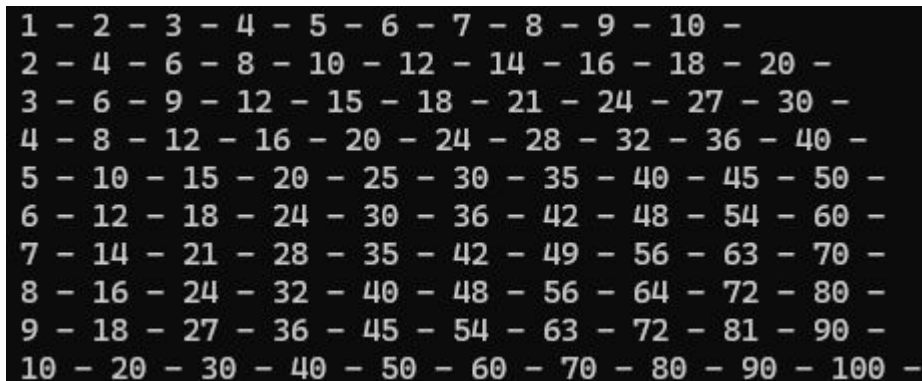
```
29 }
30
31 return 0;
32 }
```

Exercício 26

- Criar um código
 - Com array unidimensional (vetor) de 10 posições
 - Fazer um loop que vai executar 10 vezes
 - Pedir para o usuário digitar um número e guardar essa informação no array
 - Fazer um loop para percorrer o array e para cada número
 - Imprimir o número
 - Se o número for primo, imprimir a informação “eh primo” na frente do número
 - Pular de linha
- Enviar o exercício pelo Moodle
 - Conteúdo: arquivo .c

Exercício 27

- Criar um código
 - Com um array bidimensional (matriz) 10x10
 - Percorrer o array bidimensional preenchendo cada posição com o resultado da tabuada do 1 até do 10 (como a imagem)
 - Percorrer o array bidimensional e imprimir (como a imagem)
 - Pedir para o usuário digitar um número
 - Percorrer o array bidimensional verificando qual tabuada tem o resultado digitado pelo usuário
 - Quando encontrar imprimir a conta
 - Ex.: Digitou 20
 - $2 \times 10 = 20$
 - $4 \times 5 = 20$
 - $5 \times 4 = 20$
 - $10 \times 2 = 20$
- Enviar o exercício pelo Moodle
 - Conteúdo: arquivo .c



1	-	2	-	3	-	4	-	5	-	6	-	7	-	8	-	9	-	10	-
2	-	4	-	6	-	8	-	10	-	12	-	14	-	16	-	18	-	20	-
3	-	6	-	9	-	12	-	15	-	18	-	21	-	24	-	27	-	30	-
4	-	8	-	12	-	16	-	20	-	24	-	28	-	32	-	36	-	40	-
5	-	10	-	15	-	20	-	25	-	30	-	35	-	40	-	45	-	50	-
6	-	12	-	18	-	24	-	30	-	36	-	42	-	48	-	54	-	60	-
7	-	14	-	21	-	28	-	35	-	42	-	49	-	56	-	63	-	70	-
8	-	16	-	24	-	32	-	40	-	48	-	56	-	64	-	72	-	80	-
9	-	18	-	27	-	36	-	45	-	54	-	63	-	72	-	81	-	90	-
10	-	20	-	30	-	40	-	50	-	60	-	70	-	80	-	90	-	100	-

Perguntas?

Obrigado!

Até a próxima aula!