

Vetores e Matrizes

Prof. Andrei Braga
Prof. Geomar Schreiner

Sumário

- Vetores
- Matrizes
- Exercícios

Sumário

- **Vetores**
- Matrizes
- Exercícios

Vetores

- Um vetor é uma área de alocação contígua de memória
 - Ou seja, quando declaramos um vetor o programa vai alocar uma área contígua de memória baseada no tamanho do vetor
- Em C o vetor possui apenas um tipo de dado, todos os elementos ou posições do vetor armazenam o mesmo tipo de dado
- Como declarar um vetor:

```
3  int main(){
4
5      int vetor[10];
6      float vet[5000];
7
8      return 0;
9  }
```

Vetores

- Um vetor é uma área de alocação contígua de memória
 - Ou seja, quando declaramos um vetor o programa vai alocar uma área contígua de memória baseada no tamanho do vetor
- Em C o vetor possui apenas um tipo de dado, todos os elementos ou posições do vetor armazenam o mesmo tipo de dado
- Como declarar um vetor:

```
3  int m
4
5  int vetor[10];
6  float vet[5000];
7
8  return 0;
9  }
```

Tipo de
dado

Vetores

- Um vetor é uma área de alocação contígua de memória
 - Ou seja, quando declaramos um vetor o programa vai alocar uma área contígua de memória baseada no tamanho do vetor
- Em C o vetor possui apenas um tipo de dado, todos os elementos ou posições do vetor armazenam o mesmo tipo de dado
- Como declarar um vetor

```
3  int main()  
4  
5  int vetor[10];  
6  float vet[5000];  
7  
8  return 0;  
9  }
```

Nome do
vetor

Vetores

- Um vetor é uma área de alocação contígua de memória
 - Ou seja, quando declaramos um vetor o programa vai alocar uma área contígua de memória baseada no tamanho do vetor
- Em C o vetor possui apenas um tipo de dado, todos os elementos ou posições do vetor armazenam o mesmo tipo de dado

- Como declarar um

```
3  int main(){  
4  
5      int vetor[10];  
6      float vet[5000];  
7  
8      return 0;  
9  }
```

Tamanho, ou
número de
posições

Vetores

- Em C a contagem do índice inicia de 0
 - O primeiro elemento do vetor é sempre o elemento na posição 0
- Acesso é feito da mesma forma que em Python

```
3  int main(){  
4      int vetor[4];  
5  
6      vetor[0] = 1;  
7      vetor[1] = 1;  
8      vetor[2] = 1;  
9      vetor[3] = 1;  
10  
11     return 0;  
12 }
```


Vetores

- Em C a contagem do índice inicia de 0
 - O primeiro elemento do vetor é sempre o elemento na posição 0
- Acesso é feito da mesma forma que em Python

```
3  int main(  
4      int vet  
5  
6      vetor[0] = 1;  
7      vetor[1] = 1;  
8      vetor[2] = 1;  
9      vetor[3] = 1;  
10  
11     return 0;  
12 }
```

Acesso a uma
posição do
vetor.

Vetores

- O C não valida se a posição acessada de um vetor é válida, porém não há garantias nos dados fora do intervalo declarado
 - Você pode acessar a posição 11 de um vetor de tamanho 10, mas nesta área haverá lixo.
- Um vetor pode ser inicializado das seguintes formas

```
3  int main(){
4      int vetor[] = {1,2,3,4};
5
6      int vetor[4] = {1,2,3,4};
7
8      return 0;
9  }
```

Vetores

1. Considerando o vetor = $\{1,2,3,3,4,6,6,7,5,7,1,3\}$
 - a. Faça um código que imprima o vetor na tela
2. Crie um programa que lê 5 valores inteiros e, em seguida, mostre na tela os valores lidos.
3. Baseado no vetor lido na questão anterior, crie um código que imprima o maior e o menor valor contidos no vetor.
4. Crie um programa que lê 6 números inteiros e armazene em um vetor. Em seguida escreva os elementos que são pares e suas respectivas posições no vetor.

Sumário

- Vetores
- **Matrizes**
- Exercícios

Matrizes

- Matrizes são vetores de duas dimensões, ou seja, um vetor onde cada uma das posições também é um vetor
- C permite criarmos matrizes de várias dimensões, porém, o mais usual é que uma matriz apenas possua duas dimensões
- Assim como um Vetor uma Matriz é um conjunto de valores de apenas um tipo
- Como declarar:

```
3  int main(){
4      int matriz[10][10];
5      float matriz2[10][10];
6
7      int vetor[1][200];
8
9      return 0;
10 }
```

Matrizes

- Matrizes são vetores de duas dimensões, ou seja, um vetor onde cada uma das posições também é um vetor
- C permite criarmos matrizes de várias dimensões, porém, o mais usual é que uma matriz apenas possua duas dimensões
- Assim como um Vetor uma Matriz é um conjunto de valores de apenas um tipo
- Como declarar:

```
3  int main()
4  {
5      int matriz[10][10];
6      float matriz2[10][10];
7
8      int vetor[1][200];
9
10     return 0;
11 }
```

Tipo de
dado

Matrizes

- Matrizes são vetores de duas dimensões, ou seja, um vetor onde cada uma das posições também é um vetor
- C permite criarmos matrizes de várias dimensões, porém, o mais usual é que uma matriz apenas possua duas dimensões
- Assim como um Vetor uma Matriz é um conjunto de valores de apenas um tipo
- Como declarar:

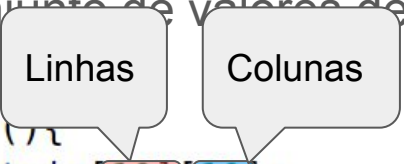
```
3  int main(){
4      int matriz[10][10];
5      float matriz2[10][10];
6
7      int vetor[1][200];
8
9      return 0;
10 }
```

Nome da
variável

Matrizes

- Matrizes são vetores de duas dimensões, ou seja, um vetor onde cada uma das posições também é um vetor
- C permite criarmos matrizes de várias dimensões, porém, o mais usual é que uma matriz apenas possua duas dimensões
- Assim como um Vetor uma Matriz é um conjunto de valores de apenas um tipo
- Como declarar:

```
3  int main()  
4      int matriz[10][10];  
5      float matriz2[10][10];  
6  
7      int vetor[1][200];  
8  
9      return 0;  
10 }
```



Matrizes

- Matrizes são vetores de duas dimensões, ou seja, um vetor onde cada uma das posições também é um vetor
- C permite criarmos matrizes de várias dimensões, porém, o mais usual é que uma matriz apenas possua duas dimensões
- Assim como um Vetor uma Matriz é um conjunto de valores de apenas um tipo
- Como declarar:

```
3  int main  
4      int ma  
5      float  
6  
7      int vetor[1][200];  
8  
9      return 0;  
10 }
```

Mesma coisa que um vetor, já que uma das dimensões é 1

Matrizes

- Uma matriz pode ser inicializada da seguinte forma:

```
3  int main(){
4      int matriz[3][3] = {
5          |   |   |   |   |   |   |   |   |   |   {00, 01, 02},
6          |   |   |   |   |   |   |   |   |   |   {10, 11, 12},
7          |   |   |   |   |   |   |   |   |   |   {20, 21, 22}
8          |   |   |   |   |   |   |   |   |   |   };
9
10     return 0;
11 }
```

Matrizes

- Exercícios
 1. Leia uma matriz 4x4 e imprima a diagonal principal.
 2. Leia uma matriz 4 x 4 e escreva a localização (linha e a coluna) do maior valor.
 3. Declare uma matriz 5 x 5. Preencha com 1 a diagonal principal e com 0 os demais elementos. Escreva ao final a matriz obtida.
 4. Faça um programa que preenche uma matriz 5 x 5 com o produto do valor da linha e da coluna de cada elemento. Em seguida, imprima na tela a matriz.