

Revisão sobre Vetores []

Estrutura da de dados
Prof. Me. João Ricardo
Semestre 2025/2

Vetores []

- Vetor é uma coleção de variáveis de mesmo tipo que compartilham o mesmo nome (identificador).
- Um vetor é uma espécie de caixa com várias divisórias para armazenar coisas (dados)
- É uma variável que pode armazenar vários valores

Vetores []

- Os vetores são definidos pelo tipo de dados que eles devem armazenar e a quantidade de posições.
- Exemplo:
 - Vetor de 8 posições para armazenar números reais
 - Vetor de 40 posições para armazenar caracteres
- Os vetores são estruturas homogêneas. Ex: um vetor de inteiros só armazena dados do tipo inteiro.

Vetores []

- Vetores
- Estrutura de dados unidimensionais
- Índice único que controla as posições
- **Sintaxe**
`<tipo><nome>[<tamanho>];`

Manipulando Vetores

- Acessando a uma posição:
 - Não é possível acessar um vetor todo
 - **Sintaxe**

<nome>[<índice>]

Lista de inicialização:preenche um vetor todo de uma só vez

<tipo> <nome> [<tam.>] = {<v1>, <v2>, ..., <vN>;

Exemplo 1

Um algoritmo que apresente 5 números inteiros e realize a media aritmética entre eles


```
#include <iostream>
```

```
int main()
```

```
{
```

```
    int v[5];
```

```
    float media;
```

```
    v[0] = 50;
```

```
    v[1] = 40;
```

```
    v[2] = 30;
```

```
    v[3] = 20;
```

```
    v[4] = 10;
```

```
    media = (v[0] + v[1] + v[2] + v[3] + v[4]) / 5.0;
```

```
    std::cout << "A media aritmetica dos numeros e: " << media << std::endl;
```

```
    return 0;
```

```
}
```

```
#include <iostream>
```

```
int main() {
```

```
    int v[5] = {10, 20, 30, 40, 50};
```

```
    int i;
```

```
    float s = 0;
```

```
    for (i = 0; i < 5; i++) {
```

```
        s += v[i];
```

```
    }
```

```
    std::cout << "Resultado: " << s / 5 << std::endl;
```

```
    return 0;
```

```
}
```


Extrapolando o Tamanho do vetor

Cuidado, pois a linguagem C++ é permissiva com relação aos índices de um vetor

0	1	2	3	4

```
#include <iostream>
```

```
int main() {
```

```
    int v[5];
```

```
    int i;
```

```
    for (i = 0; i < 6; i++) {
```

```
        std::cout << "Insira um dado: ";
```

```
        std::cin >> v[i];
```

```
    }
```

```
    std::cout << "Dados inseridos:\n";
```

```
    for (i = 0; i < 6; i++) {
```

```
        std::cout << v[i] << "\n";
```

```
    }
```

```
    return 0;
```

```
}
```


Exemplo 2

Faça um algoritmo que preencha um vetor com 25 nomes e depois imprima os nomes digitados


```
#include <iostream>
```

```
int main() {
```

```
    char nomes[5][50];
```

```
    int i;
```

```
    std::cout << "Digite os nomes:\n";
```

```
    for (i = 0; i < 5; i++) {
```

```
        std::cout << "Nome " << i + 1 << ": ";
```

```
        std::cin >> nomes[i];
```

```
    }
```

```
    std::cout << "\nNomes digitados:\n";
```

```
    for (i = 0; i < 5; i++) {
```

```
        std::cout << nomes[i] << "\n";
```

```
    }
```

```
    return 0;
```

```
}
```


Exemplo 3

Faça um algoritmo Preencher um vetor com 300 números reais. Considerando que nenhum dos valores será repetido. Ao final imprimir o menor valor digitado, e o maior valor digitado.

```
#include <iostream>
```

```
int main() {  
    float vetor[5];  
    int i;  
    float maior;  
    float menor;  
  
    std::cout << "Digite os dados para serem armazenados:\n";  
    for (i = 0; i < 5; i++) {  
        std::cin >> vetor[i];  
    }  
  
    if (i == 0) {  
        maior = vetor[i];  
        menor = vetor[i];  
    } else {  
        if (maior < vetor[i]) {  
            maior = vetor[i];  
        }  
        if (menor > vetor[i]) {  
            menor = vetor[i];  
        }  
    }  
}  
  
std::cout << "\nO maior numero digitado foi: " << maior << std::endl;  
std::cout << "O menor numero digitado foi: " << menor << std::endl;  
  
return 0;  
}
```


Atividades

Atividade

Um funcionário da Agencia dos Mercado livre de Sinop esta com problema para cadastrar e depois lembrar os CPFs cadastrados do usuários da plataforma. Faça um algoritmo para este funcionário onde o funcionário possa inserir 10 CPFs no sistema e posteriormente seja impresso todos os CPFs digitados


```
#include <iostream>
```

```
int main() {
```

```
    std::string cpfs[10];
```

```
    for (int i = 0; i < 10; ++i) {
```

```
        std::cout << "Digite o " << i + 1 << "º CPF: ";
```

```
        std::cin >> cpfs[i];
```

```
    }
```

```
    std::cout << "\n--- CPFs CADASTRADOS ---\n";
```

```
    for (int i = 0; i < 10; ++i) {
```

```
        std::cout << "CPF " << i + 1 << ": " << cpfs[i] << std::endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Atividade

Faça um algoritmo que seja solicitado o valor da tabela FIPE de 10 carros e depois imprima seus respectivos valores


```
#include <iostream>
```

```
int main() {
```

```
    float v[10];
```

```
    int i;
```

```
    for (i = 0; i < 10; ++i) {
```

```
        std::cout << "Digite o valor da tabela FIPE do carro " << i + 1 << ": ";
```

```
        std::cin >> v[i];
```

```
    }
```

```
    // Exibindo os valores inseridos
```

```
    std::cout << "\nOs valores da tabela FIPE dos carros inseridos são:\n";
```

```
    for (i = 0; i < 10; ++i) {
```

```
        std::cout << "Carro " << i + 1 << ": R$ " << v[i] << std::endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Atividade

Escreva um algoritmo que leia dois vetores de 10 posições e faça a soma dos elementos de mesmo índice, colocando o resultado em um terceiro vetor. Mostre o vetor resultante.


```
#include <iostream>
```

```
int main() {
```

```
    int vetor1[10];
```

```
    int vetor2[10];
```

```
    int vetorResultado[10];
```

```
    int i;
```

```
    // Leitura do primeiro vetor
```

```
    std::cout << "Digite os 10 elementos do primeiro vetor:\n";
```

```
    for (i = 0; i < 10; i++) {
```

```
        std::cout << "Posicao " << i << ": ";
```

```
        std::cin >> vetor1[i];
```

```
    }
```

```
    // Leitura do segundo vetor
```

```
    std::cout << "\nDigite os 10 elementos do segundo vetor:\n";
```

```
    for (i = 0; i < 10; i++) {
```

```
        std::cout << "Posicao " << i << ": ";
```

```
        std::cin >> vetor2[i];
```

```
    }
```

```
    // Soma dos elementos de mesmo índice
```

```
    for (i = 0; i < 10; i++) {
```

```
        vetorResultado[i] = vetor1[i] + vetor2[i];
```

```
    }
```

```
    // Exibindo o vetor resultante
```

```
    std::cout << "\nVetor resultante da soma dos dois vetores:\n";
```

```
    for (i = 0; i < 10; i++) {
```

```
        std::cout << "Posicao " << i << ": " << vetorResultado[i] << std::endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Atividade

Desenvolva um algoritmo para preencher um vetor com 20 números reais. Ao final imprimir a soma dos números digitados e a média dos valores digitados


```
#include <iostream>
```

```
int main() {
```

```
    float vetor[20];
```

```
    float soma = 0, media;
```

```
    int i;
```

```
    // Preenchendo o vetor com 20 números reais
```

```
    std::cout << "Digite 20 numeros reais:\n";
```

```
    for (i = 0; i < 20; i++) {
```

```
        std::cout << "Numero " << i + 1 << ": ";
```

```
        std::cin >> vetor[i];
```

```
        // Somando os valores enquanto são inseridos
```

```
        soma += vetor[i];
```

```
    }
```

```
    // Calculando a média
```

```
    media = soma / 20;
```

```
    // Exibindo os resultados
```

```
    std::cout << "\nSoma dos numeros digitados: " << soma << std::endl;
```

```
    std::cout << "Media dos valores digitados: " << media << std::endl;
```

```
    return 0;
```

```
}
```

Atividade

Elabore um algoritmo que receba o gabarito de uma prova de 10 questões e todas com questões objetivas de múltipla escolha (escolha única), posteriormente o usuário deve informar quais foram as alternativas foram por ele marcadas na prova. Considerando que cada questão vale 1,00 pontos por acertos, o algoritmo deve informar se ele foi aprovado ou não na prova onde a nota mínima para ser aprovado é 5,00 pontos.


```
#include <iostream>
```

```
int main() {
```

```
    char gabarito[10];
```

```
    char respostasUsuario[10];
```

```
    int pontos = 0;
```

```
    int i;
```

```
    // Digitar o gabarito
```

```
    std::cout << "Digite o gabarito da prova de multipla escolha:\n";
```

```
    for (i = 0; i < 10; ++i) {
```

```
        std::cout << "Questao " << i + 1 << ": ";
```

```
        std::cin >> gabarito[i];
```

```
    }
```

```
    // Digitar as respostas do usuário e calcular os pontos
```

```
    std::cout << "\nDigite as respostas marcadas por voce:\n";
```

```
    for (i = 0; i < 10; ++i) {
```

```
        std::cout << "Questao " << i + 1 << ": ";
```

```
        std::cin >> respostasUsuario[i];
```

```
        if (respostasUsuario[i] == gabarito[i]) {
```

```
            pontos++;
```

```
        }
```

```
    }
```

```
    float nota = pontos;
```

```
    // Exibir a pontuação
```

```
    std::cout << "\nPontuacao obtida: " << nota << std::endl;
```

```
    // Verificar aprovação
```

```
    if (nota >= 5.0) {
```

```
        std::cout << "Parabens! Voce foi aprovado na prova.\n";
```

```
    } else {
```

```
        std::cout << "Infelizmente, voce nao atingiu a pontuacao
```

```
        minima para aprovacao.\n";
```

```
    }
```

```
    return 0;
```

```
}
```