

Introdução a Vetores e Matrizes

Eldrey Galindo



Lembre-me:

- ✓ Chamada está disponível no Classroom
- ✓ Aula está sendo gravada e estará disponível para os alunos que solicitarem na secretaria.
- ✓ Abra a câmera



Vetores

Até agora sempre utilizamos variáveis definidas a partir de tipos básicos de dados.

- Cada variável só é capaz de armazenar um dado por vez.

Existem situações em que é necessário armazenar uma grande quantidade de dados na memória ao mesmo tempo.

- Esta quantidade torna inviável a criação de variáveis para cada um dos dados a ser armazenado.

Para estes casos, podemos criar variáveis a partir de tipos de dados estruturados.

- Um tipo de dado estruturado é aquele formado pela junção ou combinação de tipos básicos, definindo apenas uma variável que o represente.

Vetores

- Considere o problema de ordenação de 3 valores

```
inteiro a, b, c

leia(a)
leia(b)
leia(c)
se (a > b) {
    se (b > c) {
        escreva(a, " > ", b, " > ", c)
    } senao se (a > c) {
        escreva(a, " > ", c, " > ", b)
    } senao {
        escreva(c, " > ", a, " > ", b)
    }
} senao se (b > c) {
    se (a > c) {
        escreva(b, " > ", a, " > ", c)
    } senao {
        escreva(b, " > ", c, " > ", a)
    }
} senao {
    escreva(c, " > ", b, " > ", a)
}
```

Vetores

Agora considere o mesmo problema, porém, para ordenar de forma decrescente 10 números inteiros

- Este problema apresenta 3.628.800 resultados diferentes
- Desconforto de trabalhar com 10 nomes de variáveis diferentes
- É necessário o emprego de variáveis do tipo vetor para a resolução deste problema.

Vetores

Vetor (ou **array**) é uma estrutura de dados composta por uma quantidade determinada de elementos de um mesmo tipo primitivo.

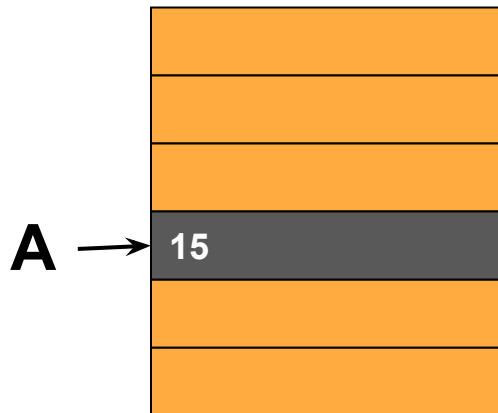
Como armazena sempre dados do mesmo tipo primitivo, diz-se que vetores são estruturas de dados **homogêneas**.

Vetores

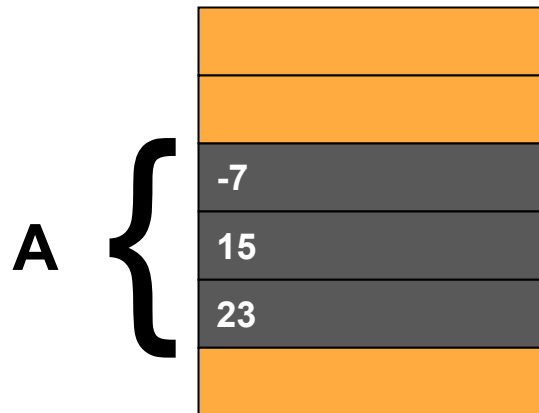
- Ao apresentar o conceito de variável, dizemos que ela funciona como uma caixa na memória do computador, capaz de armazenar um único dado de um determinado tipo.
- Por analogia, podemos imaginar um vetor como um conjunto de variáveis agrupadas sob um mesmo nome, todas de um mesmo tipo.
 - O vetor passa a ser um conjunto de caixinhas, contendo dados de mesma natureza e que estejam agrupados.

Vetores

- Armazenamento em memória



Variável simples



Vetor

Vetores

- Para referenciar o conteúdo de uma variável simples, basta indicar o nome que foi definido.
- Para vetores, somente o nome não é suficiente. Além do nome, preciso indicar a posição no vetor onde está o dado que queremos acessar/utilizar.
 - Esta posição é denominada **índice**.
 - Os vetores são também chamados de **variáveis indexadas unidimensionais**.

Vetores

- Sintaxe para declaração de um vetor:

`<tipo> <nome_vetor>[<tamanho>]`

ou

`<tipo> <nome_vetor>[] = {<valor>, <valor>, ...}`

- Sintaxe para referência de um elemento do vetor:

`<nome_vetor>[<posição>]`

- Suponha que é necessário armazenar 100 preços de produtos de uma loja.

```
real precos[100]

para(inteiro i = 0; i < 100; i++) {
    escreva(precos[i])
}
```

Vetores

Observações

- É necessário validar acessos a índices inexistentes, evitando erros no sistema;
- Apenas variáveis inteiras podem ser utilizadas como índices de um vetor.
- Cada elemento do vetor deve ser tratado como uma variável independente
- Todas as operações que podem ser realizadas sobre variáveis de tipos básicos, podem ser aplicadas a posições de um vetor.
 - Ex.: leia(preco[10]), escreva(preco[2]), preco_final[1] = lucro[1] + preco_fabrica[1]

Vetores

Exemplo:

- Uma prova de química foi feita por um grupo de 20 alunos. Faça um algoritmo para ler as notas obtidas pelos alunos, e depois exibir um relatório de notas iguais ou superiores a 7,5 no seguinte formato:

Notas boas:

7.8

9.4

10.0

7.7

8.5

7.5

9.1

8.3

Vetores

Solução:

- Este exemplo é uma aplicação simples de vetores.
- Existe uma repetição embutida, pois é necessário ler as 20 notas.
- Poderíamos solucionar este problema sem a utilização de vetores, mas perderíamos as notas dos alunos, o que é uma má ideia.
- Temos então quer ler todas as notas, e em seguida varrer todo o vetor, decidindo quais devem ser impressas.

Vetores

Como fica o algoritmo:

```
funcao inicio()
{
    real nota[20]

    para(inteiro i=0; i < 20; i++) {
        escreva("Informe a ", i+1, "a nota: ")
        leia(nota[i])
    }

    limpa()
    escreva("Notas boas")
    para(inteiro i=0; i < 20; i++) {
        se(nota[i] >= 7.5) {
            escreva("\n", nota[i])
        }
    }
}
```

Vetores

Exemplo:

- Faça um algoritmo para ler no máximo 30 números reais informados pelo usuário, e exibir depois a média desses números. Considere que o valor -99 encerra a entrada dos dados.

Vetores

Solução

- Podemos dividir este algoritmo em três etapas:
 - Ler os números;
 - Calcular a média;
 - Escrever a média.
- Devemos criar um vetor de 30 posições para armazenar os números.
 - Mas o usuário pode fornecer menos que 30 números, e aí?

Vetores

Solução (cont.):

- O fato de nem todas as posições do vetor terem sido utilizadas não é um problema, desde que saibamos quantas foram utilizadas.
- A leitura das notas não poderá ser realizada com o comando **para**, e sim com o comando **enquanto** ou **faca-enquanto**.
- O número -99 é um sinalizador de término de leitura, logo não deve ser armazenado no vetor.

Vetores

Algoritmo final

```
real entrada, numeros[30], soma, media
inteiro contador = 0

// Ler os numeros de entrada
escreva("Informe um número, ou -99 para encerrar: ")
leia(entrada)
enquanto(contador < 30 e entrada != -99) {
    numeros[contador] = entrada
    escreva("Informe um número, ou -99 para encerrar: ")
    leia(entrada)
    contador++
}

// Calcular a media
soma = 0.0
para(inteiro i=0; i < contador; i++) {
    soma += numeros[i]
}
media = soma / contador

escreva("A média dos ", contador, " números informados é ", media)
```

Vetores

Exercícios:

- Num concurso público, um candidato respondeu a uma avaliação com 80 questões de múltipla escolha, onde cada questão tinha respostas de A até D. Faça um algoritmo para ler o gabarito da prova e as respostas do aluno, informando quantas questões ele acertou.

Vetores

Exercícios:

- Faça um algoritmo para ler 50 valores reais e armazenar em um vetor. Modifique o vetor de modo que os valores das posições ímpares sejam aumentados em 5%, e os das posições pares sejam aumentados em 2%. Imprima depois o vetor resultante.

Vetores

Exercícios:

- Faça um algoritmo para ler dois vetores de números inteiros, cada um com 30 posições. Crie um terceiro vetor onde cada valor é a soma dos valores contidos nas posições respectivas dos vetores originais. Imprima depois os três vetores.

Matrizes

- Vimos que vetores são utilizados para armazenar dados de uma única “fileira”.
- Porém, existem situações em que a natureza dos dados exige uma forma de armazenamento em mais de uma dimensão.
- Para estas situações, existem as matrizes, que podem ser vistas como vetores que possuem mais de uma dimensão (normalmente duas).
- Por ser bidimensional, uma matriz representa uma tabela de valores colocados em linhas e colunas.
 - Para identificar um valor é necessário informar a **linha** e a **coluna**
 - Conhecidas como **tabelas**, ou **variáveis indexadas bidimensionais**
 - Um **vetor** pode ser visto como uma **matriz com uma única linha**.

Matrizes

- Sintaxe para declaração de uma matriz:

`<tipo> <nome_matriz>[<tamanho_x>][<tamanho_y>]`

ou

`<tipo> <nome_matriz>[][] = {{<valor>,<valor>,....},{<valor>,<valor>,....}}`

- Sintaxe para referência de um elemento:

`<nome_matriz>[<posição_x>][<posição_y>]`

- Exemplo de declaração e acesso

```
inteiro matriz[2][2] = {{15,22},{10,11}}

para(inteiro i=0; i < 2; i++) {
    para(inteiro j=0; j < 2; j++) {
        escreva(matriz[i][j])
    }
}
```

Matrizes

Exemplo

- Um distribuidor de refrigerantes vende seu produto em todo o país. Em cada trimestre do ano passado ele vendeu uma certa quantidade de garrafas em cada região do Brasil. Faça um algoritmo para ler as quantidades vendidas e escrever a quantidade total vendida em todo o país.

Matrizes

Solução

- Pelo enunciado do problema, vimos que existem 20 dados de entrada, pois temos 4 trimestres no ano e cada trimestre teve uma venda para cada uma das cinco regiões do Brasil.
- O enunciado do problema também nos leva a concluir que devemos utilizar uma estrutura matricial ao invés de vetorial.
 - Temos vendas para regiões por trimestre.

Matrizes

Solução (cont.)

- A tabela abaixo mostra uma representação possível para os dados do problema:

	1 - Norte	2 - Nordeste	3 - Sul	4 - Sudeste	5 – Centro-Oeste
1º Trimestre	150	1150	900	1500	400
2º Trimestre	180	980	1100	2300	650
3º Trimestre	140	1000	950	2100	500
4º Trimestre	210	1600	1400	2600	840

Matrizes

Então o algoritmo fica assim:

```
inteiro vendas[4][5], soma

// Leitura dos dados de vendas
para(inteiro i=0; i < 4; i++) {
    para(inteiro j=0; j < 5; j++) {
        escreva("Informe a quantidade do ", i+1, "º trimestre e ", j+1, "ª região: ")
        leia(vendas[i][j])
    }
}

// Somar quantidades
soma = 0
para(inteiro i=0; i < 4; i++) {
    para(inteiro j=0; j < 5; j++) {
        soma += vendas[i][j]
    }
}

escreva("Ao todo foram vendidas ", soma, " garrafas de refrigerante")
```

Matrizes

Exercícios

- Modifique o problema anterior para que a impressão também informe o total vendido para cada região.
 - Utilize um vetor para armazenar o total vendido para cada região.
- Faça um algoritmo para ler uma matriz 3x3 real e imprimir a soma dos elementos da diagonal principal.
- Crie uma matriz 7x8 onde cada elemento é a soma dos índices de sua posição dentro da matriz.