



INSTITUTO FEDERAL
CEARÁ



Ernani Andrade Leite

ernani.@ifce.edu.br

Aula 01 – VETORES

❑ Assunto: Vetor

❑ Objetivos:

1) Definir vetores;

2) Capacitar o aluno a operar com vetores : atribuir valores, ler dados de um vetor, escrever dados em um vetor.

❑ Roteiro:

1. Introdução;

2. Sintaxe;

3. Declaração de Vetores;

4. Operação com Vetores;

5. Exercícios.

Aula 01 – VETORES

❑ Introdução

- ❑ Durante os tópicos estudados anteriormente, percebemos que o poder de construir algoritmos se tornou cada vez maior. Porém, mesmo com o domínio das técnicas anteriores, corre-se o risco de não conseguir resolver alguns problemas, pois, a quantidade de tipos de dados estipulados não é suficiente para representar toda e qualquer informação que possa surgir.

Aula 01 – VETORES

- Assim podemos usar tipos que são construídos a partir dos tipos primitivos, os quais chamamos de estruturas de dados. Trabalhando com esses novos tipos podemos agrupar várias informações, devidamente organizadas, dentro de uma mesma variável.

Valor1	Valor2	Valor3	ValorN
Posição1	Posição2	Posição3	PosiçãoN

Aula 01 – VETORES

- ❑ A estrutura de dados que iremos estudar são os vetores. Os vetores são estruturas de dados compostas de várias posições, onde em cada posição podemos colocar valor.
- ❑ Quando nós dizemos que uma variável é do tipo vetor nela podemos armazenar várias informações do mesmo tipo (**por isto os vetores são estruturas de dados homogêneas**) colocando cada informação em uma posição diferente.

Declaração de Vetor

- ❑ Antes de trabalharmos com os vetores, nós precisamos aprender como declarar uma variável do tipo vetor nos algoritmos. Para isso, usaremos a seguinte forma:

- ❑ Sintaxe

Var

$V : \text{vetor}[Li \dots Lf] \text{ de Tipo};$

- ❑ Onde:

- V : é a variável definida como do tipo VETOR
- Li : representa o limite inicial do vetor
- Lf : representa o limite final do vetor
- TIPO : representa qualquer um dos tipos básicos (inteiro, real, caractere, lógico)

Declaração de Vetor

❑ Exemplo:

Var

Nome : vetor[1..50] de caractere;

Nota : vetor[1..50] de real;

- ❑ No exemplo visto anteriormente, foi definida a variável Nome como um vetor de 50 posições, onde cada posição pode armazenar um valor do tipo caractere. Então, se nós precisássemos armazenar os nomes de 50 alunos de uma classe, poderíamos fazer isso usando somente uma variável com 50 posições. Caso não usássemos vetores, precisaríamos declarar 50 variáveis do tipo caractere.

Operação com Vetores

- ❑ Para trabalhar com o valor armazenado em uma determinada posição de um vetor, primeiramente, informamos qual vetor contém os dados desejados e depois especificamos em qual posição este se encontra.

- ❑ Vamos observar a variável Nome do tipo vetor:

Nome : vetor [1..50] de caractere;

Operação com Vetores

- Observe que o vetor Nome foi declarado como um vetor de 50 posições onde em cada posição podemos ter um valor do tipo caracter.

Carla Ana Paula Tereza
 Posição1 Posição2 Posição3 Posição50

Operação com Vetores

- ❑ Para fazer referência ao dado armazenado em uma das 50 posições do vetor utilizamos a seguinte sintaxe:

Vetor [Indice];

- ❑ Onde:
 - Vetor : é o nome do vetor que foi utilizado na definição de variáveis.
 - Indice : pode ser uma constante, expressão aritmética ou variável que represente a posição desejada.

Operação com Vetores

❑ Exemplos:

Nome[2] {faz referência a 2ª posição do vetor Nome}

Nome [2*4+1] {faz referência a 9ª posição do vetor
Nome}

Nome [10/2] {faz referência a 5ª posição do vetor
Nome}

- ❑ A atribuição de valores a uma variável do tipo vetor é igual a atribuição para as variáveis dos tipos primitivos. Mas, devemos lembrar que os vetores são variáveis com mais de uma posição. Assim precisamos informar a qual posição o valor vai ser atribuído.

EXEMPLO

❏ Exemplo:

Algoritmo Exemplo_atribuicao;

Var

Nome : vetor [1..3] de caractere;

MeuNome : caractere;

Inicio

Nome [1] := “Silvia”; {atribui a posição 1 do vetor Nome a
cadeia “Silvia”}

MeuNome := “Marcia ”;

Nome [2] := MeuNome; {atribui a posição 2 do vetor Nome o
conteúdo da variável MeuNome}

Nome[3] := MeuNome + Nome[1]; {atribui a posição 3 do vetor Nome o
conteúdo da Variável MeuNome concatenado com o valor armazenado na
1ª posição do vetor Nome}

Fim.

Atribuição de um Vetor

- ❑ Também podemos agilizar a atribuição de valores a um vetor usando a estrutura de repetição PARA FIM PARA, como no algoritmo a seguir.
- ❑ Exemplo: Calcule a tabuada de multiplicar de 1 a 10 do número 3 e armazene-a em um vetor Tabuada3.

Atribuição de um Vetor

Algoritmo Tabuada_vet;

Var

Tabuada3 : vetor[1..10] de inteiro;

Indice : Inteiro;

Inicio

Para Indice := 1 ate 10 faca

Tabuada3 [Indice] := Indice *3 ;

Fim para;

Fim.

Leitura de dados de um Vetor

- ❑ A leitura de um vetor é processada passo a passo, um elemento por vez. Assim, se quisermos armazenar em um vetor os nomes dos alunos de uma determinada classe, teremos que usar um comando LEIA para cada posição do vetor. Exemplo:

Leitura de dados de um Vetor

Algoritmo Ler_Vetor;

Var

Nome : vetor[1..3] de caractere;

Início

Leia (Nome[1]); {permite o usuário entrar com o nome para a primeira posição do vetor Nome}

Leia (Nome[2]); {permite o usuário entrar com o nome para a segunda posição do vetor Nome}

Leia (Nome[3]); {permite o usuário entrar com o nome para a terceira posição do vetor Nome}

Fim.

Leitura de dados de um Vetor

- ❑ E se tivéssemos que entrar com 100 nomes? Para esse problema não vamos usar 100 comandos Leia, mas somente um dentro da estrutura de repetição PARA FIM PARA. Exemplo:

- ❑ Algoritmo Ler_Vetor;

Var

Nome : vetor[1..100] de caractere;

Indice : Inteiro;

Inicio

Para Indice := 1 ate 100 faca

 Leia (Nome [Indice]) ;

 Fim para;

Fim.

Leitura de dados de um Vetor

- ❑ Dessa forma a cada passo do laço podemos entrar com o valor de uma determinada posição.
- ❑ Observe que o algoritmo ficou mais compacto, além de possibilitar uma mobilidade maior, pois se houver a necessidade de entrar com um número maior de nomes, basta redimensionar o vetor e mudar o valor final da instrução PARA FIM PARA.

Escrita de dados de um Vetor

- ❑ O processo de escrita de um vetor é bastante parecido com o processo de leitura, utilizando o comando ESCRIVA. Exemplo:

Algoritmo Escreve_Vetor;

Var

Nome : vetor[1..100] de caractere;

Indice : Inteiro;

Início

Para Indice := 1 ate 100 faça

Escreva (Nome [Indice]) ;

Fim para;

Escreva (“Nome na Quarta posição do vetor é”, Nome [4]) ;

Escreva (“Nome na décima posição do vetor é”, Nome [10]) ;

Fim.

Escrita de dados de um Vetor

- ❑ Para exibir todos os elementos de um vetor usaremos também a estrutura de repetição PARA FIM PARA.

- ❑ Exercício: calcule a tabuada de multiplicar de 1 a 10 do número 3 armazene-a no vetor Tabuada3 e exiba os valores desse vetor.

EXEMPLO

Algoritmo Tabuada_vet;

Var

 Tabuada3 : vetor[1..10] de inteiro;

 Indice : Inteiro;

Inicio

 Para Indice := 1 ate 10 faca

 Tabuada3 [Indice] := Indice * 3 ;

 Fim para;

 Para Indice := 1 ate 10 faca

 Escreva (Tabuada3 [Indice]) ;

 Fim para;

Fim.

Escrita de dados de um Vetor

- ❑ OBS: Não confunda o índice, que representa a posição de um elemento do vetor, com o valor armazenado nesse elemento.
- ❑ Exercício: Faça um algoritmo que leia dois vetores de 50 posições de inteiros, some-os e escreva o resultado da soma entre os dois vetores.

EXEMPLO

Algoritmo Soma_Vetor;

Var

Vetor1, Vetor2, Vet_Soma : vetor[1..50] de Inteiro;

Indice : Inteiro;

Inicio

Para Indice := 1 ate 50 faca

 Leia (Vetor1 [Indice]) ;

 Leia (Vetor2 [Indice]) ;

Fim para;

Para Indice := 1 ate 50 faca

 Vet_Soma[Indice] := Vetor1 [Indice] + Vetor2 [Indice] ;

 Escreva (Vet_Soma [Indice]) ;

Fim para;

Fim.

Exercícios

❑ Exercício1:

- 1 – Leia duas variáveis Vet1 e Vet2 do tipo inteiro, com 20 posições cada uma.
- 2 – Escreva o vetor resultante dos maiores elementos de cada posição dos vetores lidos.

❑ Exercício2:

- 1 – Leia uma variável Vet1 do tipo inteiro, com 20 posições.
- 2 – Calcule e escreva o maior e o menor elemento do vetor.

Assunto: Algoritmo Pesq_Sequencial;

Tipo

Vetx = vetor [1..10] de inteiro;

Var

I, PESQ: inteiro;

Achou: booleano;

Início

I ← 1;

Achou ← F;

Escreva('Digite valor a ser pesquisado');

Leia (PESQ);

Enquanto (I<=10) e (Achou = F) faça

Se (PESQ = Vetx[I]) então

Achou ← T;

Senão

I ← I + 1;

Fim_se;

Fim_Entº;

Se (Achou = T) então

Escreva(PESQ, ' Foi localizado na posição', I);

Senão

Escreva (PESQ, ' Não foi localizado');

Fim_se;

FIM.

Assunto: Algoritmo Pesq_Binária;

Tipo

Vetx = vetor [1..10] de inteiro;

Var

Aux, I, J, PESQ, Começo, Fim, Meio: inteiro;

Achou: booleano;

Início

Para I ← 1 até 9 faça

Para J ← I + 1 até 10 faça

Se Vetx[I] > Vetx[J] então

Aux ← Vetx[I];

Vetx[I] ← Vetx[J];

Vetx[J] ← Aux;

Fim_se;

Fim_para;

Fim_Para.

Escreva('Digite valor a ser pesquisado');

Leia (PESQ);

Começo ← 1;

Fim ← 10;

Achou ← F;

Enquanto (Começo ≤ Fim) e (Achou = F) faça

Meio ← (Começo + Fim) div 2;

Se (PESQ = Vetx[Meio]) então

Achou ← T;

Senão

Se (PESQ < Vetx[Meio]) então

Fim ← Meio - 1;

Senão

Começo ← Meio + 1;

Fim_se;

Fim_se;

Fim_Ento;

Se (Achou = T) então

Escreva(PESQ, ' Foi localizado na posição', I);

Senão

Escreva (PESQ, ' Não foi localizado');

Fim_se;

FIM.