

Aula 13 – Vetores

Programação de Computadores I
BCC 701

VETOR

Variáveis compostas: visa facilitar a manipulação de conjuntos de dados, da mesma natureza, na memória do computador.

Variáveis compostas homogêneas: correspondem a posições de memória identificadas por um mesmo nome, individualizadas por um índice e cujo conteúdo é do mesmo tipo.

LEITURA DE UM VETOR

Nós conhecemos 3 maneiras de ler um vetor:

1. Usando um **laço for** para ler “elemento a elemento”, quando sabemos o tamanho do vetor antes de iniciar a leitura.
2. Usando um **laço while para ler** “elemento a elemento”, quando não sabemos o tamanho do vetor e este depende de uma “condição de parada”
3. Quando recebemos o vetor todo de uma única vez, “em lote”. Neste caso precisamos usar a função *length()* para saber o tamanho do vetor. Neste caso temos $v = [3, 1, 7, 4]$ no início do programa, ou digitado pelo usuário.

EXERCÍCIO 1 – Leitura 1

Escrever um programa que leia um vetor de valor numéricos, elemento a elemento, cujo tamanho é informado inicialmente pelo usuário.

Posteriormente o programa deve encontrar e informar o elemento e a posição com o maior e o menor valores.

EXERCÍCIO 1.1 – Usar uma função

Escrever um programa que leia um vetor de valor numéricos, elemento a elemento, cujo tamanho é informado inicialmente pelo usuário.

Posteriormente, escreva uma função *menorvp*(vetor) que recebe um vetor numérico e devolve o elemento e a posição com o menor valores.

```
function [valor, pos] = menorvp(vetor)
...
endfunction
```

EXERCÍCIO 2 - Leitura 2

Escrever um programa que leia um vetor de valor numéricos positivos, elemento a elemento. A leitura deve ser interrompida quando o valor digitado for ≤ 0 .

Imprimir os valores pares e suas respectivas posições.

EXERCÍCIO 3 - Leitura 3

Codificar um programa SciLab que leia um vetor “por lote”, ou seja, todos os elementos de uma única vez.

$$W = [31, 53, 68 , 12 , 98.4]$$

Use a função `length(W)` para obter o tamanho do vetor.

A seguir, o programa deve calcular a norma do vetor.

$$\text{norma} = \text{sqrt}(x_1^2 + x_2^2 + \dots + x_n^2)$$

Finalmente, o valor da norma é impresso com o formato `XXX.XXX`

EXERCÍCIO 4 - Produto Escalar

Escreva uma função *prod_escalar(v1, v2)* que recebe dois vetores numéricos do mesmo tamanho, calcula e retorna o valor do produto escalar dos vetores, dado por:

$$PE = v1(1) * v2(1) + v1(2) * v2(2) + \dots + v1(n) * v2(n)$$

Codificar um programa (principal) SciLab que leia dois vetores em lote no início do programa. O programa principal deve verificar **se os dois vetores têm o mesmo tamanho** e caso seja verdade, usar a função para obter o produto escalar dos vetores e informar este valor no formato: XXX.XXX. **Caso contrário** informar que os vetores têm tamanho diferentes.

EXERCÍCIO 5 – Geração de outro vetor

Codificar um programa SciLab que leia um conjunto de n valores inteiros, digitados pelo usuário. Para tanto, ler a dimensão do vetor e fazer a leitura de seus respectivos elementos. Usar a função `entrada_valida()` para garantir que todos os valores digitados são inteiros.

A seguir, o programa principal gera um novo vetor, contendo somente os elementos ímpares

Finalmente, o programa deve imprimir os dois vetores.

EXERCÍCIO 6 – Inversão de vetor

Codificar um programa SciLab que leia um conjunto de n valores. Para isso, o programa lê a dimensão do vetor e faz a leitura de seus respectivos elementos.

A seguir, o programa modifica esse vetor, invertendo seus elementos: o primeiro passa a ser o último, o segundo o penúltimo, e assim por diante.

O programa imprime o vetor resultante.

OBS. NÃO são alocadas novas posições de memória para o vetor resultante.

EXERCÍCIO 7 – Média e elemento < média

Codificar um programa SciLab que leia um conjunto de n valores. Para isso, o programa lê a dimensão do vetor e faz a leitura de seus respectivos elementos.

A seguir, o programa determina a média dos elementos deste vetor. Também, imprime quais elementos do vetor são menores que a média (posição e elemento). O usuário é avisado se nenhum elemento é menor do que a média.

EXERCÍCIO 8 – Pesquisa em um vetor

Codificar um programa que leia um vetor “por lote”. Posteriormente o programa deve ler um valor, digitado pelo usuário, a ser pesquisado no vetor.

Procurar no vetor o valor lido.

Se encontrar, mostrar a posição onde ele se encontra, caso contrário imprimir a mensagem, “valor não encontrado no vetor”.

EXERCÍCIO 9 – “Banco de dados”

Codificar um programa SciLab que leia três vetores de dimensão n:

- 1. um para o nome dos alunos,**
- 2. outro para as notas da prova 1 e**
- 3. outro para as notas da prova 2.**

O índice “i” especifica as informações de um mesmo aluno nos três vetores.

A seguir o programa determina a média das notas de cada aluno, armazenando-a em um novo vetor.

O programa imprime o nome do aluno, suas notas e sua média.

EXERCÍCIO 10

Codificar um programa SciLab que leia dois vetores (V1 e V2) de dimensão “n”. A seguir, o programa gera um novo vetor (V3), de dimensão “2n”, com os elementos intercalados dos vetores lidos, ou seja:

$$V1 = [a \ b \ c \ d \ e]$$

$$V2 = [1 \ 2 \ 3 \ 4 \ 5]$$

$$V3 = [a \ 1 \ b \ 2 \ c \ 3 \ d \ 4 \ e \ 5]$$

O programa imprime todos os vetores.

EXERCÍCIO 11

Considere um vetor lido no início do programa por atribuição: $v = [1, 4, 5, 2, 6, 7, 8, 17]$.

Codificar um programa Scilab que crie um novo vetor a partir deste que contenha apenas os elementos pares.

Se $v = [1, 4, 5, 2, 6, 7, 8, 17]$, então o vetor w gerado pelo programa deve ser

$w = [4, 2, 6, 8]$