



Universidade Federal do Vale do São Francisco
Curso de Engenharia de Computação



Introdução a Algoritmos – Parte 06

(Baseado no Material do Prof. Marcelo Linder)

Prof. Jorge Cavalcanti

jorge.cavalcanti@univasf.edu.br

www.univasf.edu.br/~jorge.cavalcanti

www.twitter.com/jorgecav

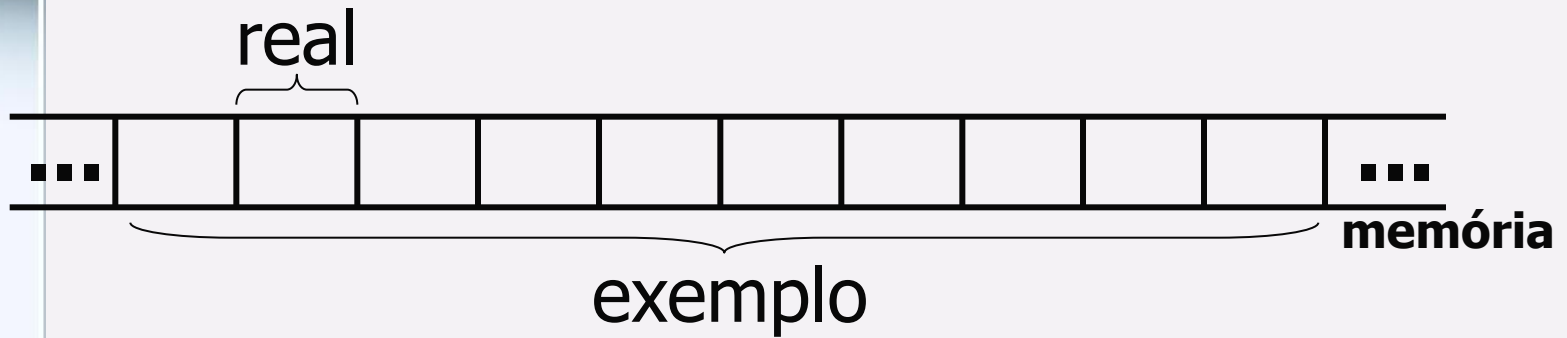
Estruturas de dados homogêneas

- As estruturas de dados homogêneas permitem agrupar diversas informações dentro de uma mesma variável. Este agrupamento ocorrerá obedecendo sempre ao mesmo tipo de dado, e é por esta razão que estas estruturas são chamadas **homogêneas**.
- A utilização deste tipo de estrutura de dados recebe diversos nomes, como: variáveis indexadas, variáveis compostas, variáveis subscritas, arranjos, vetores, matrizes, tabelas em memória ou *arrays*.
- Os nomes mais usados e que utilizaremos para estruturas homogêneas são: matrizes (genérico) e vetores (matriz de uma linha e várias colunas).
- Nesta disciplina, utilizaremos o conceito de **VETOR**.
- É importante ressaltar que vetores de qualquer dimensão são caracterizados por terem todos os seus elementos pertencentes ao mesmo tipo de dado.
- A forma geral para se declarar um vetor unidimensional é:
nome_do_vetor: vetor[menor_indice..Maior_indice] de
tipo_dos_elementos

Estruturas de dados homogêneas

Exemplo 1:

exemplo: vetor [1..10] de real



Estruturas de dados homogêneas

- Em função de um vetor se tratar de um arranjo de elementos torna-se necessária uma forma de acessar individualmente cada elemento. A indexação possibilita tal acesso.
- A especificação do intervalo dos índices além de definir o número de elemento indica quais serão os valores dos índices utilizados para acessar cada elemento.
- No exemplo anterior, os dados serão indexados de 1 a 10. Para acessá-los vamos escrever:

exemplo[**1**]

exemplo[**2**]

...

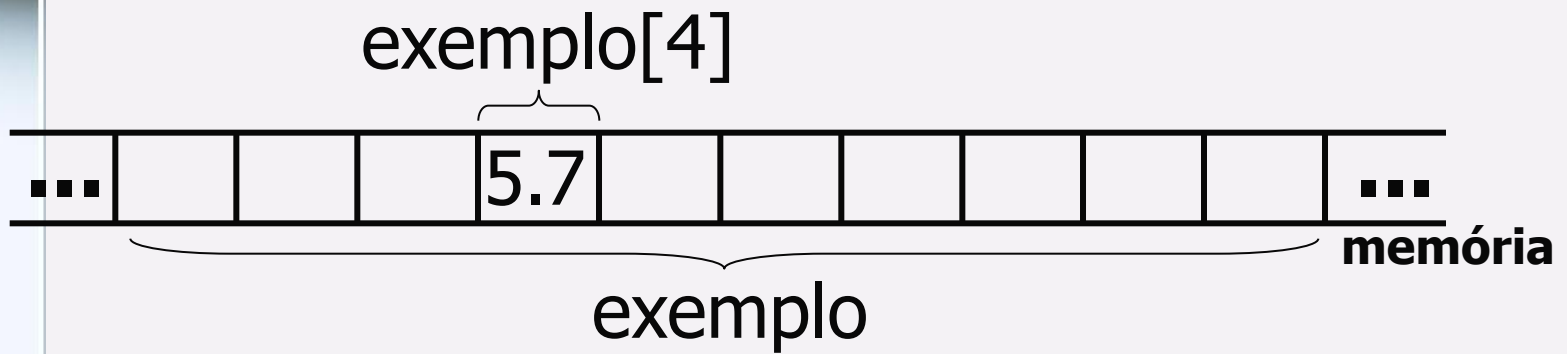
exemplo[**10**]

OBS: Tenha cuidado para não confundir o índice com o elemento. O índice é o endereço, enquanto o elemento é o conteúdo armazenado em um determinado endereço.

Estruturas de dados homogêneas

Exemplo:

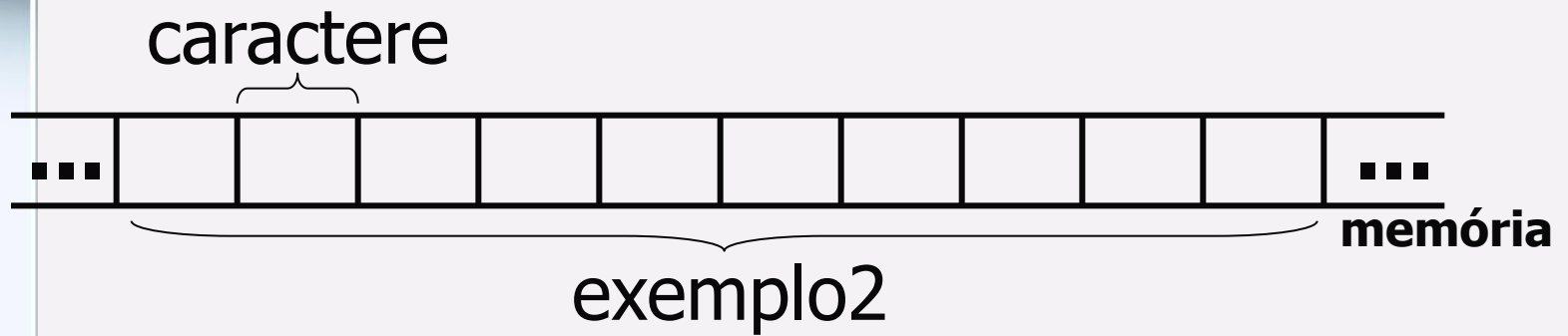
exemplo[4] <- 5.7



Estruturas de dados homogêneas

Exemplo 2:

exemplo2: vetor [4..13] de caractere



Estruturas de dados homogêneas

- No último exemplo apresentado, os dados serão indexados de 4 a 13. Para acessá-los vamos escrever:

exemplo2[4]

exemplo2[5]

...

exemplo2[13]

Observação: Não é permitida a utilização de índices negativos!

- Para solicitar a entrada de valores do vetor pelo usuário, deve-se colocar:

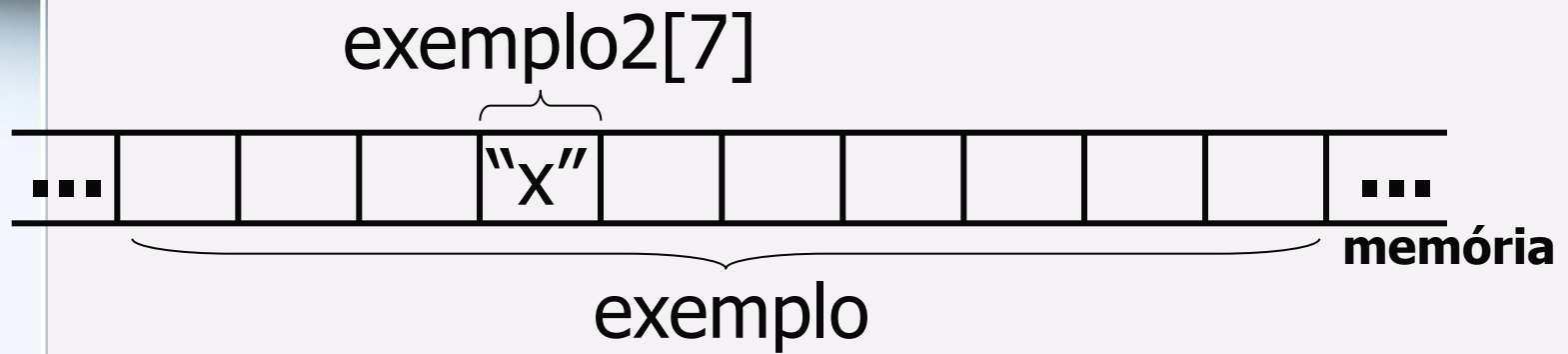
leia nome_do_vetor([índice])

- Uma observação importante a ser feita é a utilização da construção **para** a fim de efetuar a operação de leitura repetidas vezes, em cada uma delas lendo um determinado componente do vetor.
- De fato esta construção é muito comum quando se opera com vetores, devido à necessidade de se realizar uma mesma operação com os diversos componentes dos mesmos.

Estruturas de dados homogêneas

Exemplo 2:

exemplo2[7] <- "x"



Estruturas de dados homogêneas

Exercício:

- Construa um algoritmo que declare um vetor de inteiros com 12 elementos e o inicialize com números fornecidos pelo usuário, através da entrada padrão.

algoritmo "exercício vetor"

var vet:vetor [1..12] de inteiro

 i:inteiro

início

 para i de 1 ate 12 faça

 escreva ("Entre com vetor[",i,"]: ")

 leia (vet[i])

 fimpara

fimalgoritmo

Estruturas de dados homogêneas

Outra forma:

- Construa um algoritmo que declare um vetor de inteiros com 12 elementos e o inicialize com números fornecidos pelo usuário, através da entrada padrão.

algoritmo "exercício vetor"

var vet:vetor [0..11] de inteiro

i:inteiro

inicio

para i de 0 ate 11 faca

escreva ("Entre com vetor[",i+1,"]: ")

leia (vet[i])

fimpara

fimalgoritmo

Estruturas de dados homogêneas

■ Escrita de Dados de um Vetor

■ A escrita de um vetor obedece à mesma sintaxe da instrução primitiva de saída de dados e também vale lembrar que, além do nome do vetor, deve-se também especificar por meio do índice o componente a ser escrito.

escreva (<nome_do_vetor> [<índice>])

■ O algoritmo a seguir exemplifica a operação de leitura e escrita de um vetor, utilizando a construção para:

```
algoritmo "escrita_vetor"
var vet:vetor [0..10] de inteiro
    i:inteiro
inicio
    para i de 0 ate 10 faca
        escreva ("Entre com vetor[",i,"]: ")
        leia (vet[i])
    fimpara
    para i de 0 ate 10 faca
        escreva (vet[i])
    fimpara
finalgoritmo
```

Estruturas de dados homogêneas

Exercício:

- Fazer um algoritmo para ler 10 valores que devem ser armazenados em um vetor, calcular e mostrar a soma e média desses valores.

```
algoritmo "vetor_exemplo"
var vet:vetor [1..10] de real
    i:inteiro
    soma, media: real
inicio
    para i de 1 ate 10 faca
        escreva ("Entre com vetor["i,"]: ")
        leia (vet[i])
        soma <- soma + vet[i]
    fimpara
    media<-soma/10
    escreval ("Soma dos valores: ", soma)
    escreval ("Média dos valores: ", media)
finalgoritmo
```

Estruturas de dados homogêneas

Usando vetores:	Usando variáveis simples:
<p><u>Algoritmo</u></p> <pre> n : inteiro ← 10; nome : vetor [1 .. n] de caracter; ldade : vetor [1 .. n] de inteiro; x : inteiro; soma, media : real; início para x de 1 até n faça leia (Nome [x], ldade[x]); soma ← soma + ldade [x]; fim para; media ← soma / n; escreva ('Média das idades:', media); para x de 1 até n faça escreva (Nome [x]); fim para; fim.</pre>	<p><u>Algoritmo</u></p> <pre> Nome1, Nome2, Nome3, Nome4: caracter; Nome5, Nome6, Nome7, Nome8: caracter; Nome9, Nome10 : caracter; ldade1, ldade2, ldade3, ldade4, ldade5: inteiro; ldade6, ldade7, ldade8, ldade9, ldade10 : inteiro; x : inteiro; soma, media : real; início leia (Nome1, Nome2, Nome3, Nome4, Nome5); leia (Nome6, Nome7, Nome8, Nome9, Nome10); leia (ldade1, ldade2, ldade3, ldade4, ldade5); leia (ldade6, ldade7, ldade8, ldade9, ldade10); soma ← ldade1+ldade2+ldade3+ldade4+ ldade5+ldade6+ldade7+ldade8+ldade9+ ldade10; media ← soma / 10; escreva ('Média das idades:', media); escreva(Nome1, Nome2, Nome3, Nome4); escreva(Nome5, Nome6, Nome7, Nome8); escreva(Nome9, Nome10); fim.</pre>

Estruturas de dados homogêneas

Exercício:

- Elabore um algoritmo, com base no exercício anterior, que declare um vetor de inteiros com 12 elementos, o inicialize, com números fornecidos pelo usuário através da entrada padrão, e que através de uma pesquisa nos elementos do vetor, retorne na saída padrão os elementos de menor e maior valor, respectivamente.



Estruturas de dados homogêneas

Exercício: Ler 8 elementos de um vetor A. Construir um vetor B de mesma dimensão com os elementos do vetor A multiplicados por 3. Dessa forma o elemento B[1] deve ser implicado pelo elemento A[1] * 3, o elemento B[2] deve ser implicado pelo elemento A[2] * 3. No final você deverá apresentar o vetor B.

```
algoritmo "vetor_exemplo2"
var
a:vetor[1..8] de inteiro
b:vetor[1..8] de inteiro
posicao: inteiro
inicio
para posicao de 1 ate 8 faca
escreva("DIGITE O ", posicao, "º VALOR: ")
leia(a[posicao])
b[posicao] <- a[posicao] * 3
fimpara
para posicao de 1 ate 8 faca
escreval(a[posicao], " * 3 = ", b[posicao])
fimpara
finalgoritmo
```

Estruturas de dados homogêneas

Vetores Multidimensionais

- Em determinados problemas são utilizados estruturas de vetores com mais de uma dimensão. O mais comum é a matriz de duas dimensões por se relacionar com a utilização de tabelas.
 - Matrizes com mais de duas dimensões são utilizadas com menos frequência.
- Uma matriz de duas dimensões é atribuída pela instrução vetor:
nome: vetor [linhas, colunas] de tipo de dado
- Uma matriz de duas dimensões está sempre fazendo menção a linhas e colunas e é exemplificada a seguir por seu nome de uma matriz chamada **tabela**, com um tamanho de quatro linhas (de 1 a 4) e 5 colunas (de 1 a 5), ou seja, é uma matriz de 4 por 5 (4 x 5).

tabela: vetor [1..4, 1..5] de inteiro

Estruturas de dados homogêneas

Vetores Multidimensionais

- Isso significa que podem ser armazenados em **tabela** até 20 elementos.

1,1	1,2	1,3	1,4	1,5
2,1	2,2	2,3	2,4	2,5
3,1	3,2	3,3	3,4	3,5
4,1	4,2	4,3	4,4	4,5

- A leitura de uma matriz de duas dimensões é processada passo a passo, um elemento por vez, sendo utilizada a instrução "leia" seguida da variável mais os seus índices.

leia(tabela[1,3])

- O processo de escrita é bastante parecido com o processo de leitura de seus elementos.

escreva (tabela [2,2])

Estruturas de dados homogêneas

■ Vetores Multidimensionais - Exemplos

tabela: **vetor**[1..5,1..3] de **inteiro**

medias: **vetor**[1..10,1..4] de **real**

- Vetores com N dimensões podem ser criados a partir da sintaxe:

**nome_do_vetor : vetor [menor_indice_d1..
maior_indice_d1, menor_indice_d2..
maior_indice_d2, ..., menor_indice_dn..
maior_indice_dn] de tipo_dos_elementos**

- Para fazer referência a uma posição particular ou elemento no array, especificamos o nome do array e o número da posição do elemento no array.

Estruturas de dados homogêneas

Vetores Multidimensionais (continuação)

- Um importante aspecto a ser considerado é que na manipulação de uma matriz do tipo vetor é utilizada uma única instrução de laço (para, enquanto ou repita).
- No caso de matrizes com mais dimensões, deve ser utilizado o número de laços relativos ao tamanho de sua dimensão.
- Desta forma uma matriz de duas dimensões deve ser controlada com dois laços, de três dimensões por 3 laços e assim por diante.

Estruturas de dados homogêneas

Vetores Multidimensionais (continuação)

- Considere que eu uma turma existam 10 alunos e para cada aluno deverá ser armazenado as médias referentes às quatro unidades.
- Sendo assim, para resolver esse problema poderíamos utilizar uma matriz de dimensão 10 x 4 (10 linhas representando os 10 alunos e 4 colunas representando cada uma das notas).
- Para realizar a leitura dos dados, será necessário realizar um mecanismo para ler as 4 médias de cada aluno. Para isso usaremos duas estruturas de repetição conforme exemplo abaixo:

```
para i de 1 ate 10 faca
  para j de 1 ate 4 faca
    leia(notas[i,j])
  fimpara
fimpara
```

Estruturas de dados homogêneas

Exemplo 1:

- O algoritmo abaixo declara uma matriz 3x4 de inteiros e a inicializa com valores fornecidos pelo usuário.

Algoritmo "exemplo matriz"

var

matriz: vetor [1..3,1..4] de inteiro

i, j: inteiro

inicio

para i de 1 ate 3 faca

para j de 1 ate 4 faca

escreva ("Entre com matriz[",i, ", ",j, "]: ")

leia (matriz[i,j])

fimpara

fimpara

fimalgoritmo

Estruturas de dados homogêneas

Exemplo 2:

- Ler dois vetores A e B, cada um com uma dimensão para 5 elementos. Construir em seguida uma matriz C de duas dimensões, de forma que a primeira coluna da matriz C deve ser formada pelos elementos da matriz A multiplicados por 2 e a segunda coluna deve ser formada pelos elementos da matriz B subtraídos de 5. Apresentar ao final a matriz C.



Estruturas de dados homogêneas

```
algoritmo "soma_vetor_igual_matriz"
var
a: vetor[1..5] de inteiro
b: vetor[1..5] de inteiro
c: vetor [1..5, 1..2] de inteiro
i,j: inteiro

inicio
escreval("LENDO OS VALORES DO VETOR A...")
para i de 1 ate 5 faca
    escreva("a[",i,"] = ")
    leia(a[i])
fimpara

escreval("LENDO OS VALORES DO VETOR B...")
para i de 1 ate 5 faca
    escreva("b[",i,"] = ")
    leia(b[i])
fimpara
```

Estruturas de dados homogêneas

```
//preenchendo a primeira e segunda linha da matriz C
para i de 1 ate 5 faca
    c[i,1] <- a[i] * 2
    c[i,2] <- b[i] - 5
fimpara
escreval("ESCREVENDO OS NOVOS VALORES DA MATRIZ ...")
para i de 1 ate 5 faca
    para j de 1 ate 2 faca
        escreval("c[",i,",",j,"] = ", c[i,j])
    fimpara
fimpara
finalgoritmo
```


Estruturas de dados homogêneas

Exercício 1

Construa um algoritmo que declare uma matriz 7x4 de números reais, a inicialize com valores fornecidos pelo usuário através da entrada padrão e a apresente na saída padrão com o layout a seguir:

	x.xx	x.xx	...	x.xx
	10	10		
	x.xx	x.xx	...	x.xx

...

algoritmo "exercício 1 matriz"

var

matriz: vetor [1..7,1..4] de real

i, j: inteiro

inicio

para i de 1 ate 7 faca

para j de 1 ate 4 faca

escreva ("Entre com matriz[" ,i, " ,",j,"]= ")

leia (matriz[i,j])

fimpara

fimpara

para i de 1 ate 7 faca

escreva ("|")

para j de 1 ate 4 faca

escreva (matriz[i,j]:10:2)

fimpara

escreval ("|")

fimpara

fimalgoritmo

Estruturas de dados homogêneas

Exercício 2:

- Com base no exercício 1 anterior, faça um algoritmo que declare duas matrizes 2×3 de reais, as inicialize e efetue a soma das duas matrizes retornando a matriz resultante na saída padrão com o layout dado.

Estruturas de dados homogêneas

Exercício:

Escreva um algoritmo com uma função que indique se há ocorrências de um determinado valor dentro de um vetor de 100 elementos inteiros. Caso haja a ocorrência do valor, a função deverá retornar a posição, dentro do vetor, em que o elemento se encontra. Caso o valor não exista, o valor -1 deve ser retornado.

Estruturas de dados homogêneas

```
algoritmo "Localiza_no_vetor_exemplo1"
var
vet100: vetor[1..10] de inteiro // tam 10 para facilitar a entrada de dados
num: inteiro
posicao: inteiro
indice: inteiro

funcao busca(n:inteiro): inteiro
var
i: inteiro
inicio
i<-1
enquanto ((vet100[i] <> n) e (i<=10)) faca
    i<- i+1
    se (i<=10) entao
        retorne (i)
    senao
        retorne (-1)
fimse
fimenquanto
fimfuncao
```

Estruturas de dados homogêneas

```
//continuacao

inicio
para posicao de 1 ate 10 faca // carrega o vetor
    escreva("DIGITE O ", posicao, "º VALOR: ")
    leia(vet100[posicao])
fimpara
    escreval("DIGITE O NUMERO A SER PESQUISADO: ")
    leia(num)
    indice := busca(num)
se (indice <> -1) entao
    escreva ("A posição no vetor é: " , indice)
senao
    escreva ("numero não encontrado")
fimse

finalgoritmo
```

Estruturas de dados homogêneas

Outra Forma:

algoritmo " Localiza_no_vetor_exemplo2"

var

i,k, vb : inteiro

v : vetor[1..10] de inteiro

achou : logico

procedimento preenchevetor

// Preenche o vetor com numeros aleatorios entre 0 e 100

var j : inteiro

inicio

para j de 1 ate 10 faca

 v[j] <- randi(100)

 escreval(j:5,v[j]:5)

fimpara

Fimprocedimento

Estruturas de dados homogêneas

Outra Forma continuação:

```
// algoritmo " Localiza_no_vetor_exemplo2"
```

```
inicio
```

```
  preenchevetor
```

```
  escreva("Entre com o valor de busca (ESC termina) :")
```

```
  leia(vb)
```

```
  // Busca Seqüencial
```

```
  i <- 1
```

```
  repita
```

```
    se v[i] = vb entao
```

```
      achou <- verdadeiro
```

```
    senao
```

```
      i <- i+1
```

```
    fimse
```

```
  ate (achou=verdadeiro) ou (i>10)
```

```
  se achou entao
```

```
    escreval("Achei ", vb , " na posição ", i)
```

```
  senao
```

```
    escreval("Nao achei.")
```

```
  fimse
```

```
finalgoritmo
```


Estruturas de dados homogêneas

Exercício:

Elabore um algoritmo que manipule uma matriz de inteiros. O algoritmo deve possuir um módulo para inicializar a matriz com informações fornecidas pelo usuário, outro para apresentá-la na saída padrão, com o layout adequado, e por fim, um módulo que retorne os dois maiores valores contidos na matriz. O número de elementos contidos na matriz é fornecido pelo usuário, sendo que o número de elementos em uma dimensão não pode exceder 20. Os módulos devem ser utilizados de forma satisfatória pelo algoritmo.

Estruturas de dados homogêneas

Exercício:

Escreva um algoritmo com uma função que receba um vetor de tamanho definido pelo usuário e gere um novo vetor com as entradas invertidas (ex.: $\text{vet1}[1] = \text{vet2}[10], \dots$)

Estruturas de dados homogêneas

Exercício:

Desenvolva um algoritmo que leia um vetor com 10 posições inteiras e o coloque em ordem crescente, utilizando a seguinte estratégia de ordenação:

- 1 - Selecione o elemento do vetor de 10 posições que apresenta o menor valor;
- 2 - Troque este elemento pelo primeiro;
- 3 - Repita estas operações, envolvendo agora apenas os 9 elementos restantes (selecionando o de menor valor com a segunda posição).
- 4 - Depois os 8 elementos (trocando o de menor valor pela terceira posição), depois os 7. os 6 e assim por diante, até restar um único elemento, o maior deles.

Estruturas de dados homogêneas

Exercício:

algoritmo "ordenacao simples"

var

v:vetor[1..10]de inteiro //vetor de entrada de dados

i,j,k:inteiro //índice

aux: inteiro //variável auxiliar para troca

procedimento preenchevetor

// Preenche o vetor com numeros aleatorios entre 0 e 100

var j : inteiro

inicio

escreval ("Vetor Original")

para j de 1 ate 10 faca

v[j] <- randi(100)

escreval(j:5,v[j]:5)

fimpara

fimprocedimento

Estruturas de dados homogêneas

Exercício Cont:

Início

preenchevetor

//ordenação do vetor

para i de 1 ate 9 faca

k <- i

aux <- v[i] //Guarda dados para verificação de menor número

para j de i + 1 ate 10 passo 1 faca

se v[j] < aux entao // Faz verificações dos números para encontrar o menor.

k <- j //var auxilia trocas de valores abaixo

aux <- v[k] // atribui menor valor em aux

fimse

fimpara

v[k]<- v[i] //limpa v[i]

v[i]<- aux //atribui menor valor na variável de saida v[i]

fimpara

//laço para mostrar o vetor já ordenado.

escreval ("Vetor Ordenado")

para i de 1 ate 10 faca

escreval (i:6,v[i]:6)

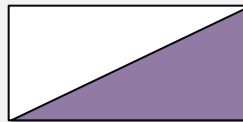
fimpara

fimalgoritmo

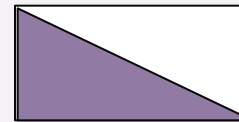
Estruturas de dados homogêneas

Exercício 1- Matrizes:

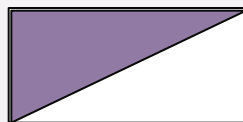
- a) Escreva um algoritmo que escreva a diagonal principal de uma matriz 4×4 .
- b) Faça um algoritmo que imprima as seguintes partes de uma matriz quadrada 5×5 , a partir da escolha do usuário.



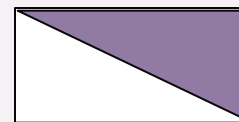
Primeiro Triângulo



Terceiro Triângulo



Segundo Triângulo



Quarto Triângulo

Estruturas de dados homogêneas

Exercício 2- Matrizes:

O tempo de voo entre várias cidades está colocado na tabela abaixo. Note que a diagonal principal é nula pois representa que origem = destino.

- a) Construa um algoritmo que leia a tabela e informe ao usuário o tempo para percorrer duas cidades por ele fornecida, até o momento em que ele fornecer duas cidades iguais (origem e destino).
- b) Desenvolva um algoritmo que permita ao usuário informar várias cidades, até inserir uma cidade "0" e que mostre o tempo total para cumprir todo o percurso entre as cidades fornecidas.

Cidades	1	2	3	4	5	6
1		2	11	6	15	11
2	2		7	12	4	2
3	11	7		11	8	3
4	6	12	11		10	2
5	15	4	8	10		5
6	11	2	3	2	5	

Estruturas de dados homogêneas

Exercício 3- Matrizes:

Escreva um algoritmo que preencha uma matriz 5 X 5 de inteiros e escreva:

- a) A soma dos números ímpares fornecidos;
- b) A soma de cada uma das 5 colunas;
- c) A soma de cada uma das 5 linhas.

Exercício 4- Matrizes:

Escrever um algoritmo para ler uma matriz (5 X 4) contendo valores inteiros. Depois encontrar e mostrar o menor valor contido na matriz e sua posição.