ALGORTIMOS E PROGRAMAÇÃO DE COMPUTADORES

AULA 6

Estruturas de Dados Homogêneas

Vandor Roberto Vilardi Rissoli

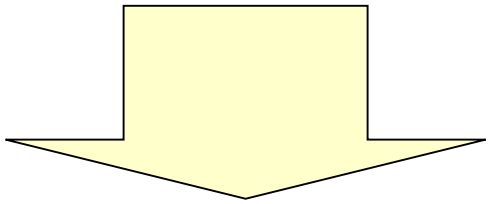
APRESENTAÇÃO

- Estrutura de Dados Homogênea
- Vetores
- Passagem de Vetor por Parâmetros
- Ordenação de Dados
- Referências

Estrutura de Dados Homogênea

SUPONHA A SITUAÇÃO!!

- Como fazer um programa que leia as notas de vários alunos, calcule a média e determine quais alunos tiveram nota superior à média. O número de alunos não é conhecido e será informado pelo usuário, mas se sabe que é menor que 50.



Estruturas de Dados

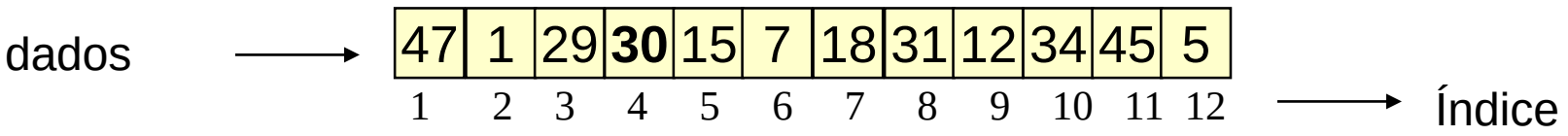
- HOMOGÊNEAS
  - Vetores e matrizes
- HETERÔGÊNEAS
  - Registros

Estrutura de Dados Homogênea

Variáveis Compostas Unidimensionais - VETORES

- Um vetor é uma variável que pode armazenar mais de um valor:
- As características dos vetores são:
  - Contém vários valores (número definido)
  - Todos valores são do mesmo tipo de dado
  - O vetor possui um único nome (identificador)
  - Cada valor do conjunto é acessível independentemente, de acordo com o seu ***índice*** ou ***posição*** na estrutura

Estrutura de Dados Homogênea



- Estrutura chamada de VETOR com 12 valores inteiros
- O nome (identificador) do vetor é "dados"
- O valor do índice (posição) 4 deste vetor guarda **30**

➔ Vetores são chamados de "variáveis compostas unidimensionais homogêneas":

- "composta" porque é criada uma única variável que tem a capacidade de armazenar vários valores
- "unidimensional" porque um vetor só tem variação em uma dimensão (só varia o índice da linha no exemplo acima)
- "homogênea" porque o vetor armazena somente valores de um mesmo tipo de dado

Vetores

- Declaração de variável convencional:

  <tipo de dado> <identificador> ;

  Exemplos: real nota;

  caracter sexo;

- Declaração de vetores:

  <tipo de dado> <identificador>[<tamanho>];

  onde: tipo de dado = tipo de dado escalar (primitivo)

  identificador = nome do vetor (regras de nome)

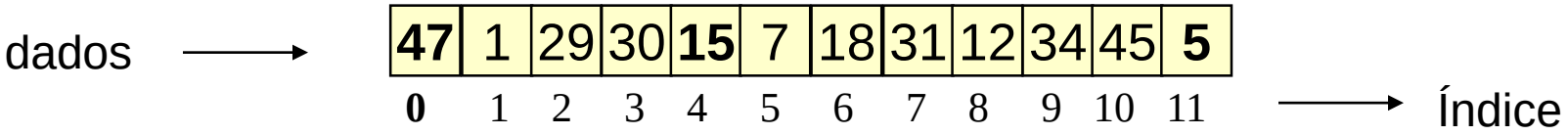
  tamanho = valor inteiro que define a quantidade de elementos do vetor (de 0 a tamanho -1)

  A declaração do exemplo anterior seria:

  inteiro dados[12];

Vetores

No exemplo anterior o vetor **dados** pode armazenar 12 valores inteiros, variando o seu índice no intervalo definido na declaração do vetor, ou seja, **de 0 até 11**. Sendo assim, este vetor tem a capacidade de armazenar, em uma única variável, até doze valores.



⇒ Declaração - inteiro dados [12];

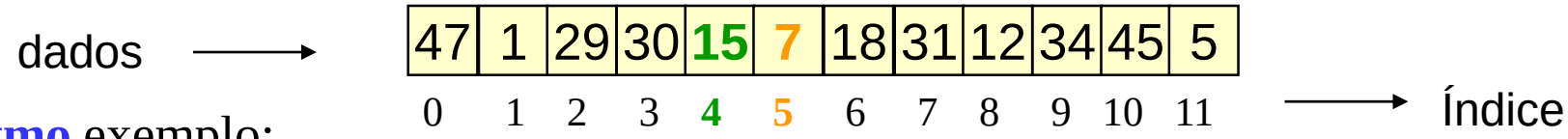
⇒ O tamanho do vetor é de 12 elementos (ou posições)

⇒ O quinto elemento está na posição 4 e guarda o valor **15**

⇒ A posição inicial do vetor é zero e possui o valor **47**, enquanto sua última posição é 11 e possui o valor **5**

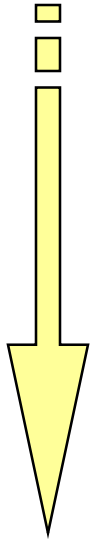
Vetores

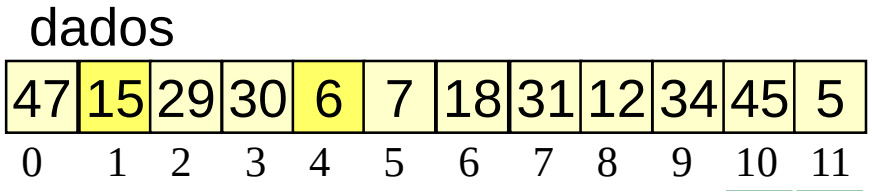
- Para acessar um elemento do vetor, se usa o nome do vetor e o índice do elemento desejado entre colchetes, por exemplo: o valor de dados[5] é 7 e de dados[4] é 15.




```
algoritmo exemplo;
// Síntese
//   objetivo: manipular dados do vetor
//   entrada: ...
//   saída: ...
principal
  // Declarações
      inteiro aux;
      inteiro dados [12];
  // Instruções
       :::
      aux =  4;
      dados[1] = dados[aux];
      dados[aux] = dados[2*(aux-1)+2]/2;
       :::
fimPrincipal
```

tamanho do vetor





Vetores

- Escrever um algoritmo que declare um vetor de reais e leia as notas de 30 alunos.

```
algoritmo notas;
// Síntese
//    Objetivo: armazenar as notas de 30 alunos de uma turma
//    Entrada: 30 notas
//    Saída: notas dos alunos
principal
   // Declarações
      inteiro aux;
      inteiro quantidade;    // CONSTANTE
      real notas[30];
   // Instruções
      quantidade = 30;
      para (aux de 0 ate quantidade-1 passo 1) faca
         escreva("Nota do aluno ", aux + 1, " = ");
         leia(notas[aux]);
      fimPara
fimPrincipal
```

- Agora, altere este algoritmo para validar a nota por função
- Apresente todas as notas da turma que foram cadastras.
- Complete o algoritmo, calculando a **média da turma** e determine quais e quantos alunos tiveram nota igual ou superior à média 7.

Exercícios de Fixação

1) Altere o algoritmo anterior considerando que não se conhece quantos alunos esta turma tem (menos que 50). O número de alunos será informado pelo usuário e validado.

2) Faça um algoritmo que calcule e escreva o somatório dos valores armazenados numa variável composta unidimensional, chamada **dados,** de 70 elementos numéricos a serem lidos no dispositivo de entrada padrão.

Exercícios de Fixação

3)Faça um algoritmo que leia **até 30** letras e as escreva na ordem inversa (ou contrária) da que foram lidas no vetor.

4)Crie um algoritmo que leia um vetor de 80 elementos numéricos e verifique se existem elementos iguais a 120. Se existirem escrever quantas vezes eles aparecem e quais as posições em que eles estão armazenados no vetor.

5) Elabore um algoritmo que calcule e escreva o somatório dos valores armazenados numa variável composta unidimensional de 100 elementos inteiros a serem lidos do usuário, onde a operação de soma só poderá ser realizada pela função ***somaTudo*** e a média destes valores será calculada e mostrada pelo procedimento ***mediaTotal***.

Atenção! O estudo de vetores de caractere é especializado – (String).

Passagem de Vetor por Parâmetros

1- Passagem de um elemento:

```
algoritmo umElemento;
// Síntese
//   Objetivo: mostrar passagem de um elemento do vetor como parâmetro
//   Entrada: notas de 3 alunos
//   Saída: notas dos alunos
principal
      // Declarações
         inteiro auxiliar;
         real notas[3];
      // Instruções
         para (auxiliar de 0 ate 2 passo 1) faca
              escreva("Informe a nota do aluno ", auxiliar + 1 , " = ");
              leia(notas[auxiliar]);
         fimPara
         limpaTela();
         escreval("Mostra as notas");
         para (auxiliar de 0 ate 2 passo 1) faca
               mostraNota(notas[auxiliar], auxiliar + 1);
         fimPara
fimPrincipal
```

Passagem de Vetor por Parâmetros

```
// continuação da passagem de parâmetros de um elemento

    // Subprograma

      // Objetivo: mostrar a nota de um aluno
      // Parâmetros: nota de um aluno
      // Retorno : nenhum

       procedimento mostraNota(real nota, inteiro aluno)
            escreval(" Nota do aluno ",aluno, " : ",nota);
       fimProcedimento
```

Passagem de Vetor por Parâmetros

2- Passagem de uma referência a todos elementos

```
algoritmo todosElementos;
// Síntese
//    Objetivo: mostrar passagem da referência aos elementos de um vetor
//    Entrada: notas de 3 alunos
//    Saída: notas dos alunos
principal
      // Declarações
        inteiro auxiliar;
        real notas[3];
      // Instruções
        leNotas(notas, 3);
        limpaTela();
        mostraNotas(notas, 3);
fimPrincipal
```

Passagem da referência aos elementos do vetor declarado no algoritmo principal

Passagem de Vetor por Parâmetros

// continuação da passagem de uma referência a todos elementos

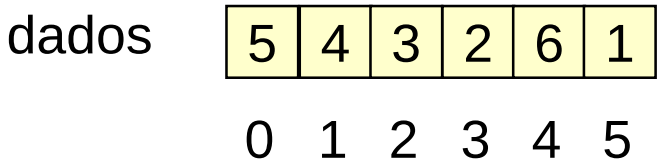
```
// Objetivo: mostrar as notas de todos os alunos
// Parâmetros: referência as notas dos alunos e  quantidade de alunos
// Retorno : nenhum
procedimento mostraNotas(real notaAlunos[], inteiro qtdeAlunos)
      // Declarações
        inteiro auxiliar;
      // Instruções
        para (auxiliar de 0 ate qtdeAlunos-1 passo 1) faca
              escreval(" Nota do aluno ",auxiliar + 1, " : ", notaAlunos[auxiliar]);
        fimPara
fimProcedimento

// Objetivo: ler as notas de todos os alunos
// Parâmetros: referência as notas dos alunos quantidade de alunos
// Retorno : nenhum
procedimento leNotas(real notaAlunos[], inteiro qtdeAlunos)
      // Declarações
        inteiro auxiliar;
     // Instruções
        para (auxiliar de 0 ate qtdeAlunos -1 passo 1) faca
              escreva("Nota do aluno ", auxiliar+1 , " = ");
              leia(notaAlunos[auxiliar]);
        fimPara
fimProcedimento
```

Exercícios de Fixação

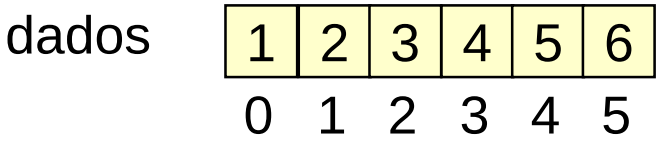
6) Fazer um algoritmo que leia a matrícula e o salário dos funcionários de uma empresa (máximo de 100 funcionários definidos em uma constante). Após a leitura de todos os dados, informe em uma janela limpa os dados lidos e o maior e menor salário dos funcionários cadastrados. O maior e menor salário deverão ser obtidos, cada um, por uma função. Sabe-se ainda que **não existem matrículas repetidas** na empresa, pois estas são **CHAVES** que identificam um funcionário unicamente na empresa.

Ordenação de Dados

- Seja o vetor



- Como se pode ordenar os seus valores?



Ordenação de Dados

- Existem diferentes lógicas com diferentes algoritmos de classificação (ou ordenação).
- Um deles consiste da comparação de cada elemento com todos os elementos subsequentes. O elemento será trocado de posição com um outro elemento, dependendo dele ser menor ou maior que o mesmo, conforme a ordenação desejada:
  - CRESCENTE
  - DECRESCENTE

Ordenação de Dados

Suponha a situação onde o usuário informou os valores armazenados no vetor abaixo e é necessário apresentá-los em ordem crescente. Como esta ordenação poderia ser realizada?

Verifica 1º elemento

5	4	3	2	6	1
4	5	3	2	6	1
3	5	4	2	6	1
2	5	4	3	6	1
2	5	4	3	6	1
1	5	4	3	6	2

Verifica 2º elemento

1	4	5	3	6	2
1	3	5	4	6	2
1	3	5	4	6	2
1	2	5	4	6	3

Verifica 3º elemento

1	2	4	5	6	3
1	2	4	5	6	3
1	2	3	5	6	4

Verifica 4º elemento

1	2	3	5	6	4
1	2	3	4	6	5

Verifica 5º elemento

1	2	3	4	5	6

Ordenação de Dados

- Elabore o algoritmo para ordenar seis valores por troca.

```
algoritmo ordenacao;
// Síntese
//    Objetivo: ordenar 6 números
//    Entrada: 6 números
//    Saída: números ordenados
principal
  // Declarações
     real numeros [6], troca;
     inteiro aux, indice;
  // Instruções
     // leitura dos dados
     para (aux de 0 ate 5 passo 1) faca
       escreva(aux+1, " Numero= ");
       leia(numeros[aux]);
     fimPara
  // processo de ordenação
     para (aux de 0 ate 4 passo 1) faca
        para (indice de aux+1 ate 5 passo 1) faca
           se (numeros[aux] > numeros[indice]) entao
              troca = numeros[indice];
              numeros[indice] = numeros[aux];
              numeros[aux] = troca;
           fimSe
        fimPara
     fimPara
    // mostra os valores ordenados
     limpaTela();
     para (aux de 0 ate 5 passo 1) faca
         escreval(aux+1," Numero= ",numeros[aux]);
     fimPara
fimPrincipal
```

Exercícios de Fixação

7) Desenvolva um algoritmo que cadastre os 100 preços (real) diferentes de CD em uma loja. Após este cadastro, o algoritmo deverá possibilitar a execução de uma das duas funções para aumento dos preços cadastrados, sendo a primeira de 10% para todos os CDs, enquanto a segunda aumenta o percentual informado pelo vendedor para todos os CDs. Será o vendedor que escolherá o tipo de aumento desejado e o percentual livre tem que ser maior que zero.

8) Fazer um algoritmo que leia **até 30** caracteres e os ordene em ordem decrescente. É considerado que A < B < C < ... (a ordem alfabética também é reconhecida pelo computador).

Exercícios de Fixação

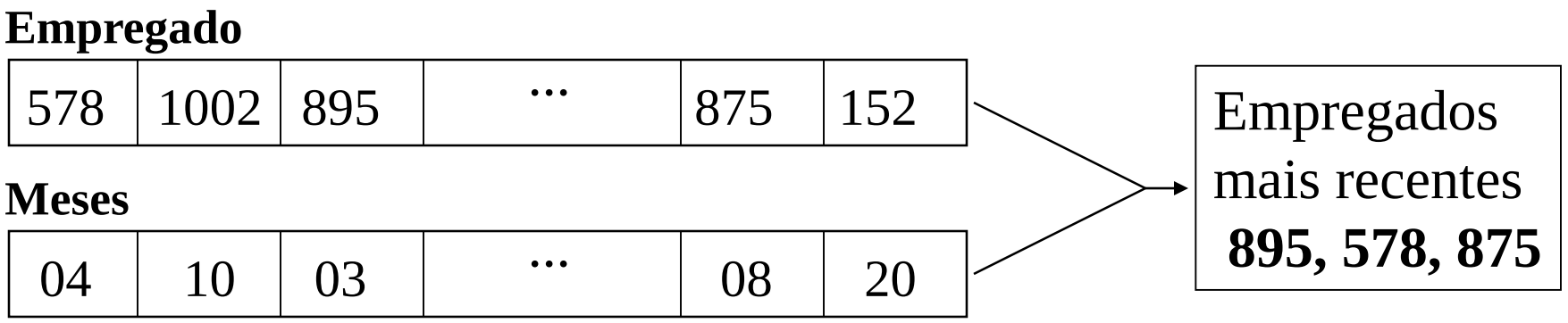
09) Faça um algoritmo que leia a idade de **até 80** pessoas e apresente a média entre todas, além de identificar a mais velha e a posição em que ela se encontra no vetor.

10) Escrever um algoritmo que solicite e leia **letra por letra** o nome completo de uma pessoa. O caractere sustenido "#" indica o fim do nome. Depois de lê-lo apresente todo o nome informado, sem o **#.** Garanta que o limite máximo do vetor, que foi definido em sua declaração, não será superado em hipótese alguma.

11) Elabore um algoritmo que armazene a **matrícula inteira** de um aluno e sua respectiva **média final de nota** (real) guardando-os em vetores. Mostre a maior e a menor média entre todos os alunos cadastrados, usando um subalgoritmo para encontrar a maior e outro para encontrar a menor média. Por fim, acione outra função para calcular a média das médias das notas cadastradas.

Exercícios de Fixação

12) Uma grande empresa deseja saber quais os três empregados mais recentes. Fazer um algoritmo para ler um número indeterminado de informações (máximo de 100) contendo a matrícula funcional do empregado e o número de meses de trabalho deste empregado. Mostre os três empregados mais recentes e garanta em sua solução que não existirão empregados admitidos no mesmo mês. A matrícula funcional igual a zero ‘0’ (zero) encerra a leitura. Observe o exemplo abaixo:



Exercícios de Fixação

13) Fazer um algoritmo que:

a) leia uma frase de 80 caracteres, incluindo os espaços

b) conte quantos espaços em branco existem na frase

c) conte quantas vezes a letra ‘a’ ou ‘A’ aparece

d) conte quantas vezes ocorre um mesmo par de letras na frase e quais são elas;

e) apresente o que foi calculado nos itens **b**, **c** e **d**

14) Elabore um algoritmo que leia um conjunto de valores inteiros correspondentes a 80 notas, variando de 0 a 10, dos alunos de uma **turma** (vários alunos). Calcule a frequência de cada nota e apresente uma tabela contendo os valores das notas e suas respectivas frequências.

Referências de Criação para Apoio ao Estudo

Material para Consulta e Apoio ao Conteúdo

- FARRER, H. *et al*. Algoritmos Estruturados, Editora LTC, 3ª . edição, 1999.
  - Capítulo 2
- MANZANO, J.; OLIVEIRA, J. Algoritmos, Lógica para desenvolvimento de programação, Editora Ética, 1996.
  - Capítulo 6, 7 e 8
- Universidade de Brasília (UnB Gama)
  - https://sae.unb.br/cae/conteudo/unbfga

    (escolha a disciplina **Algoritmo Prog Computadores**)

Estrutura Homogênea em C

- Declaração de vetores em **Algoritmo**:

  <tipo de dado> <identificador>[<tamanho>];

  - Exemplo: real notas[50];

- Declaração de vetores na **Linguagem C**:

  <tipo de dado> <identificador>[<tamanho>];

  - onde: tipo de dado = tipo de dado da linguagem C
    identificador = nome do vetor (regras)
    tamanho = valor inteiro que define quantidade de elementos do vetor que variará de 0 até (tamanho -1)
  - Exemplo: float notas[50];

Estrutura Homogênea em C

- A declaração de uma **string** corresponde a um vetor especial na Linguagem C, pois exige certos cuidados

  <char> <identificador>[<tamanho>];

  - Exemplo: char nomeCompleto[70];

Cuidados com String na Linguagem C

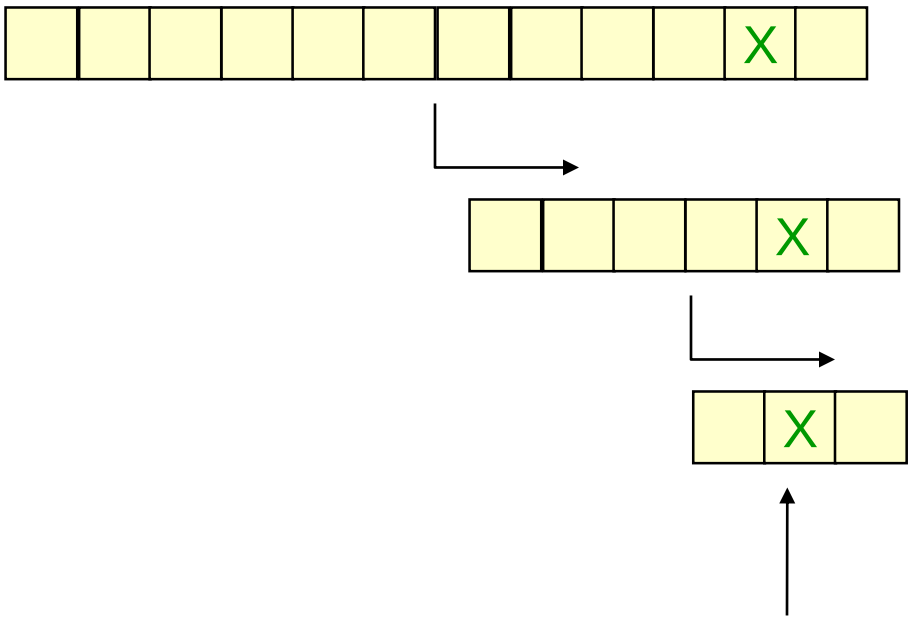
- Posição começa em zero (0) e vai até tamanho -1
- Sempre tem um terminador que ocupa o espaço de um caracter (1 byte em 1 posição da memória dentro do vetor)
- Só armazena 1 caracter em cada posição do vetor de **char**
- Terminador corresponde ao código '**\0**' que é só um **char**
- Toda **string** deve prever uma posição para guardar o terminador dentro de suas posições criadas na declaração
- Verificar se terminador cabe na string (**erro grave invasão**)

Pesquisa ou Busca

- Como pesquisar um valor em um vetor.
- No quarto (4) exercício de pesquisar valores igual a 120, realizou-se o que é chamado de **PESQUISA SEQUENCIAL,** cujas características são:
  - pesquisar todos os dados do vetor sequencialmente (um após outro) até encontrar o valor desejado
  - o número médio de comparações é N/2 onde N é o número de elementos do vetor
- Quando os elementos estão ordenados pode-se fazer outro tipo de pesquisa mais eficiente, denominada: **PESQUISA BINARIA**

Pesquisa Binária

- Procura-se o elemento X desejado dividindo o vetor em duas partes e testando em qual das duas partes ele deve estar



Exercícios de Fixação

15) Alterar o algoritmo de ordenação de caracteres para pesquisar um caractere específico anterior para realizar a Pesquisa Binária.

16) Ler uma matriz A e B, cada uma com 10 elementos. Gerar uma matriz C onde cada elemento corresponde a resultado da operação de adição entre as matrizes (A + B). Faça a pesquisa se existe algum elemento na matriz resultante (C) que seja igual a um valor fornecido pelo usuário.