Algoritmos de Programação

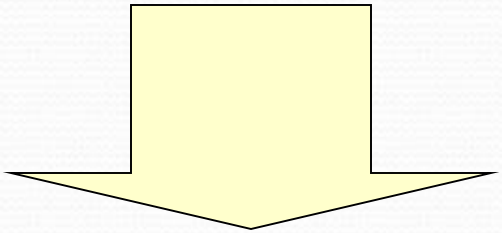
Curso: Análise e Desenvolvimento de Sistemas
Modalidade: Presencial
Professor Esp. Wesley Tschiedel
Email: wesley.tschiedel@ucb.br

Unidade 10 (Estrutura de Dados homogêneas - vetores numéricos - algoritmo)

Estrutura de Dados Homogênea

SITUAÇÃO!!

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



Estruturas de Dados

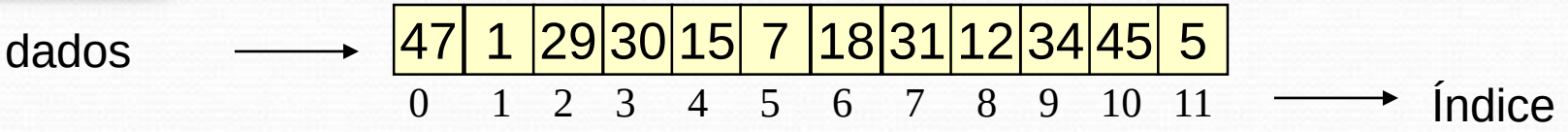
- Homogêneas
  - Vetores e matrizes
- Heterogêneas
  - Registros

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 dados
  - O vetor possui um único nome
  - Cada valor do conjunto é acessível independentemente, de acordo com o seu ***índice***
  - ***Os índices dos vetores começam sempre em zero***

VETORES - exemplo



- Vetor com 12 valores inteiros
- O nome do vetor é "dados"
- O valor do índice (posição) 4 deste vetor = 15

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

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

VETORES - Sintaxe

- Declaração de variável convencional:

  <tipo de dado> <identificador> ;

  – Exemplo: real nota;
  texto nome;

- Declaração de vetores:

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

onde: tipo de dado = tipo de dado escalar

identificador = nome do vetor (regras de nome)

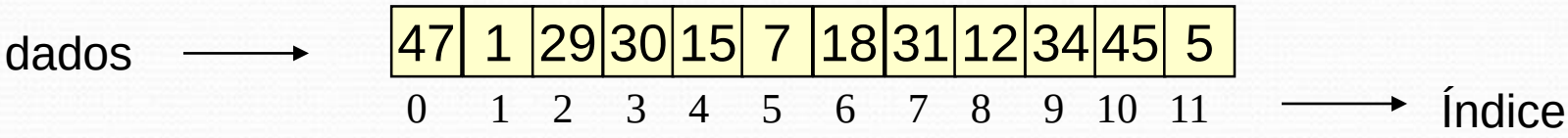
tamanho = valor inteiro que define a quantidade de elementos do vetor. O índice varia de 0 a tamanho -1

Para o exemplo anterior teria-se:

inteiro dados[12];

Estrutura de Dados Homogênea - exemplo

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



⇒ Inteiro dados[12];

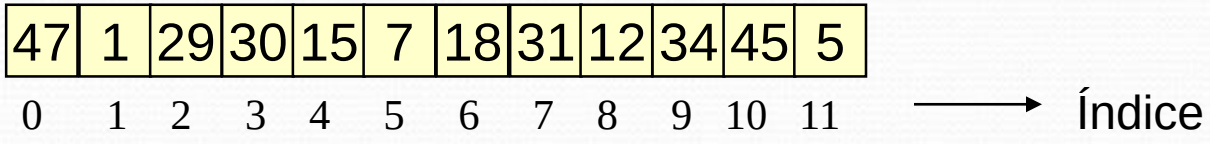
⇒ O tamanho do vetor é de 12 elementos (ou posições)

⇒ O sexto elemento (índice 5) possui o valor 7

⇒ A posição de índice 4 corresponde ao quinto elemento do vetor e possui o valor 15

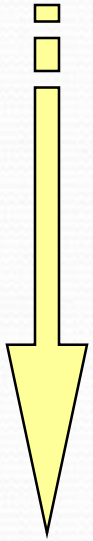
Estrutura de Dados Homogênea - acesso

- 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[4] é 15




```
algoritmo exemplo;
// Síntese
//   objetivo: ...
//   entrada: ...
//   saída: ...
principal
      // Declarações
      inteiro i;
      inteiro dados[12];
      // Instruções
      :::
      i =  4;
      dados[1] = dados[i];
      dados[i] = dados[2*i-1]/3;
      :::
fimPrincipal
```

Tamanho do vetor (índices de 0 a 11)



dados

47	15	29	30	10	7	18	31	12	34	45	5
0	1	2	3	4	5	6	7	8	9	10	11

Estrutura de Dados Homogênea - exercício

- Escrever um programa 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 auxiliar, quantidade;
    real notas[30];
    // Instruções
    quantidade = 30;   // valor constante
    para (auxiliar de 0 ate quantidade-1 passo 1) faca
        escreva("Nota do aluno ", auxiliar+1, " = ");
        leia(notas[auxiliar]);
   fimpara
fimalgoritmo
```

- Complete este algoritmo para validar a nota entre 0 a 10 por uma função

- Complete este algoritmo para apresentar todas as notas da turma.

- Complete o algoritmo para calcular a média da turma e determine quais e quantos alunos tiveram nota superior à média 7.

Passagem de Estrutura de Dados Homogênea como parâmetros para funções

1- Passagem de um elemento:

```
algoritmo umElemento;
// Síntese
//    Objetivo: mostrar a 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("Nota do aluno ", auxiliar+1 , " = ");
            leia(notas[auxiliar]);
       fimPara
       limpaTela();
```

Passagem de Estrutura de Dados Homogênea como parâmetros para funções

1- Passagem de um elemento (continuação)

```
    para (auxiliar de 0 ate 2 passo 1) faca
        mostraNota(notas[auxiliar],auxiliar+1);
     fimPara
fimPrincipal

//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 Estrutura de Dados Homogênea como parâmetros para funções

2- Passagem de uma referência aos elementos

```
algoritmo todosElementos;
// Síntese
//    Objetivo: mostrar a 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
        leiaNotas(notas,3);
        limpaTela();
        apresentaNotas(notas,3);
fimPrincipal
```

Passagem da referência aos elementos do vetor declarado no algoritmo principal

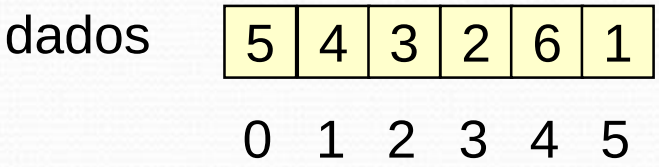
Passagem de Estrutura de Dados Homogênea como parâmetros para funções

2- Passagem de uma referência aos elementos (continuação)

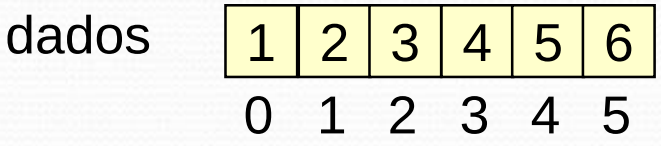
```
//Objetivo: mostrar as notas de todos os alunos
//Parâmetros: referência as notas dos alunos
//                  qtde de alunos
//Retorno : nenhum
procedimento apresentaNotas(real notaAlunos[], inteiro qtdeAlunos)
      inteiro auxiliar;
      para (auxiliar de 0 ate qtdeAlunos-1 passo 1) faca
 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

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

Ordenação dos Dados no Vetor

- Seja o vetor



- Como pode-se ordenar os seus valores?



Estrutura de Dados Homogênea - Ordenação

- 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

Estrutura de Dados Homogênea - exemplo

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

ALGORITMO DE ORDENAÇÃO

- 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," Numero= ",numeros[aux]:10:2);
    fimPara
fimPrincipal
```

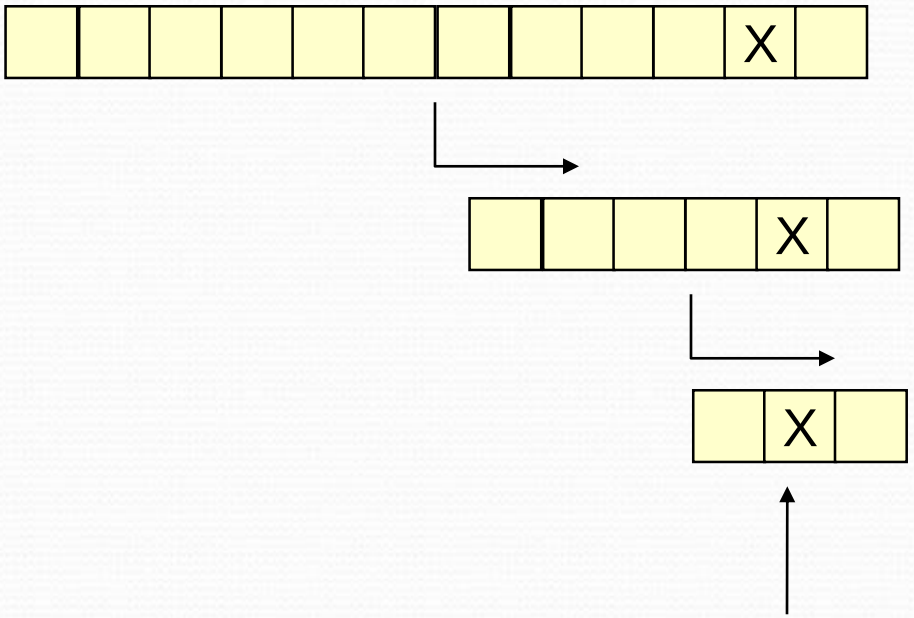
PESQUISA OU BUSCA

- 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, denominada:
  - **PESQUISA BINARIA**

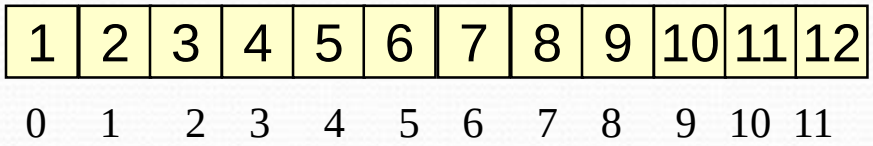
PESQUISA BINÁRIA

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



PESQUISA BINÁRIA – exemplo:

→ Procurar: **2**




```
algoritmo pesquisa;
// Síntese
//    Objetivo: pesquisar/ordenar um vetor
//    Entrada: 12 números e 1 número para pesquisa
//    Saída: localização do valor pesquisado
principal
    // Declarações
    inteiro comeco,fim,meio,busca,aux,contador,outro;
    inteiro dados [12];
     // Instruções
    // leitura de dados
     para (aux de 0 ate 11 passo 1) faca
        escreva("Informe o ",aux+1," número= ");
        leia (dados[aux]);
     fimPara

    // valor a ser pesquisado
     escreva("Qual valor de pesquisa: ");
     leia(busca);
```

continuação do exemplo

```
// Ordenação sequencial por troca
 para (aux de 0 ate 10 passo 1) faca
    para(contador de aux+1 ate 11 passo 1)faca
       se (dados[aux] > dados[contador]) entao
          outro =dados[aux];
          dados[aux] =dados[contador];
          dados[contador] =outro;
       fimSe
    fimPara
 fimPara

 // pesquisa o valor informado
 comeco = 0;
 fim = 11;
 faca
    meio = (comeco + fim) \ 2;
    se (busca < dados[meio]) entao
       fim = meio - 1;
    senao
       comeco = meio + 1;
    fimSe
 enquanto((dados[meio]!= busca) e (comeco <= fim));
       se (dados[meio] == busca) entao
          escreva (busca, " esta em ", meio+1);
       senao
          escreva ("Não existe o elemento");
       fimSe
fimPrincipal
```

Vamos Praticar!!!

Referências Bibliográficas

Básica:

- EVARISTO, J. **Aprendendo a programar: Programando em C.** Book Express, 2001.
- FARRER, H. etall. **Algoritmos Estruturados.** 3ª ed. LTC, 1999.
- MANZANO, J.; OLIVEIRA, J. **Algoritmos: Lógica para Desenvolvimento de Programação.** 6ª ed. São Paulo: Ética, 2000.

Referências Bibliográficas

Complementar:

- FORBELLONE, A. L. V. **Lógica de Programação: A construção de algoritmos e estrutura de dados.** Makron Books, 1993.
- GUIMARÃES, A.; LAGES, N. **Algoritmos e Estrutura de Dados.** LTC, 1994.
- MIZRAHI, V. V. **Treinamento em linguagem C: Módulo 2.** São Paulo: Makron Books, 1990.
- SALVETTI, D. D; BARBOSA, L. M. **Algoritmos.** São Paulo: Makron Books, 1998.
- SCHILDT, H. **C: Completo e total.** 3ª ed. São Paulo: Makron Books, 1997.