

# Vetores

Introdução à Ciência da Computação  
Rosane Minghim  
Guilherme Pimentel Telles  
Apoio na confecção: Rogério Eduardo Garcia  
Danilo Medeiros Eler

## Vetores

- Tipos Compostos
    - Conjunto de vários elementos de mesmo tipo
    - Indexáveis individual e diretamente
  - Um único nome. Ex: V
  - Um tipo. Ex: V:inteiro
- ```
v [ inteiro | inteiro | inteiro | inteiro | inteiro | inteiro | inteiro ]
```
- Armazenamento em sequência. V:inteiro[1..7]
  - Um índice.
    - Acesso direto: V[j]

## Exemplos

tipo frase = caracter[0..80]

variável  
valor: real[1..10]  
linha: frase

valor[1] ← 0.35  
linha[0] ← ‘:’

## Exemplos – índice negativo e operações

```
variável  
abcissas: real[-5..5]  
j:inteiro  
valor:real  
prox_abs:real  
  
abcissas[4] ← 10  
abcissas[-1] ← 3  
abcissas[0] ← -2  
abcissas[-2] ← 3  
j ← -5  
abcissas[j] ← 0  
abcissas[1] ← abcissas[5] + abcissas[-5]  
valor ← raiz(abcissas[-1],2)  
prox_abs ← abcissas[ truncar(abcissas[0]) ]  
Após a seqüência completa o valor de prox_abs seria 3
```

## Exemplo: média aritmética de números lidos

```
Algoritmo média  
variável  
soma, n,i: inteiro  
aux:inteiro  
média:real  
  
leia(n)  
i ← 0  
enquanto (i<n) faça  
  leia(aux)  
  soma ← soma + aux  
  i ← i + 1  
fim  
média ← soma/n  
escreva(soma,média)  
fim
```

## Exemplo: média aritmética de números lidos

E se também quiséssemos imprimir todos os números, entre aqueles lidos, que sejam maiores ou iguais a média?

```
Algoritmo média  
tipo vetor50 = inteiro[1..50]  
variável  
soma, n: inteiro  
valor: vetor50  
i:inteiro  
  
leia(n)  
soma ← 0  
para i de 1 até n passo 1 faça  
  leia(valor[i])  
  soma ← soma + valor[i]  
fim para  
  
média ← soma/n  
escreva(média)  
para i de 1 até n passo 1 faça  
  se (valor[i] >= média) então  
    escreva(valor[i])  
  fim se  
fim para  
fim
```

## Exercício

Desenvolver um programa para ler um vetor VAL de números inteiros e criar outro vetor VAL2 de mesma quantidade de elementos que VAL, onde os elementos tenham o dobro do valor dos elementos de VAL. O número máximo de elementos é 30.

## Vetores em subprogramas - Exemplo

Suponha o subprograma dado na aula anterior (subprogramas), que possui o seguinte cabeçalho:

```
calcule_pontos (questão, resposta, pontos, acerto)
```

e atualiza os pontos obtidos na prova por um candidato com base na resposta de uma questão em particular.

Desenvolver um subprograma para, dado um vetor de 10 elementos contendo a resposta a cada uma das 10 questões da prova, calcular a soma de pontos de um funcionário. Além disso, o subprograma retorna um vetor de 10 posições contendo, em cada posição, o valor *verdadeiro* se a questão está correta e o valor *falso* se a questão está errada.

## Vetores em subprogramas - Exemplo

Dois vetores são necessários: um, de entrada, para armazenar as respostas e um outro, de saída, para conter quais estão corretas.

Vamos definir tipos para eles:

```
constante N_QUESTÕES = 10
```

tipo

```
vetor_respostas = caracter[1..N_QUESTÕES]
```

```
vetor_acertos = lógico[1..N_QUESTÕES]
```

## Vetores em subprogramas - Exemplo

O subprograma recebe o vetor de respostas e fornece, como saída, além da soma total de pontos, o vetor de acertos. O cabeçalho para o subprograma é, portanto:

```
Subprograma  
calcule_resultado_prova(respostas, acertos, número_de_pontos)
```

e: respostas: vetor\_resposta {um vetor contendo, em cada posição, a resposta a uma questão da prova}

s: acertos: vetor\_acertos {um vetor contendo, em cada posição, falso se a resposta à questão associada for errada, e verdadeiro caso ela seja correta}  
número\_de\_pontos: inteiro {o número de pontos conseguido na prova}

{este subprograma utiliza o subprograma calcule\_pontos}

## Vetores em subprogramas - Exemplo

O código do subprograma deve, para cada elemento do vetor de respostas, chamar o subprograma calcule pontos. O resultado do acerto, devolvido por ele, deve ser armazenado no vetor de acertos. O código fica:

```
variável  
i: inteiro  
certo: lógico  
início  
número_de_pontos ← 0  
para i de 1 até N_QUESTÕES passo 1 faça  
  resp ← respostas[i]  
  calcule_pontos(i, resp, número_de_pontos, certo)  
  acertos[i] ← certo  
fim para  
fim
```

## Vetores em subprogramas - Exemplo

```
Subprograma calcule_resultado_prova(respostas, acertos, número_de_pontos)
```

e: respostas: vetor\_resposta {um vetor contendo, em cada posição, a resposta a uma questão da prova}

s: acertos: vetor\_acertos {um vetor contendo, em cada posição, falso se a resposta à questão associada for errada, e verdadeiro caso ela seja correta}

número\_de\_pontos: inteiro {o número de pontos conseguido na prova}

{este subprograma utiliza o subprograma calcule\_pontos}

```
variável  
i: inteiro  
início  
número_de_pontos ← 0  
Para i de 1 até N_QUESTÕES passo 1 faça  
  calcule_pontos(i, respostas[i], número_de_pontos, acertos[i])  
fim para  
fim
```

## Vetores - Inicialização

O pseudo-código admite a definição de vetores constantes, da seguinte forma:

constante  
nome:tipo = {c1,c2,...c\_max}

Onde:  
tipo é a definição de um vetor e  
c1,c2,...c\_max são constantes do tipo elementar armazenado no vetor.

Alguns exemplos:

tipo  
vet = inteiro[1..6]  
constante  
v: vet = {4,5,4,3,3,1}  
c: caracter[1..3] = {'A', 'B', 'C'}

## Vetores em subprogramas - Exercícios

1. Fazer um algoritmo principal que, utilizando o subprograma `calcule_resultado_prova`, leia do usuário as respostas para os funcionários da empresa que prestaram a prova, imprimindo, para cada um deles, a pontuação total e as o número das perguntas respondidas corretamente.
2. Alterar o algoritmo acima para, além das impressões solicitadas, imprimir o número do funcionário que tirou a maior nota.
3. Fazer um subprograma para ler do usuário um vetor genérico de números reais com, no máximo, 50 elementos.
4. Fazer um subprograma para escrever o conteúdo de um vetor genérico com, no máximo, 50 elementos.

## Vetores em subprogramas - Exercícios

1. Desenvolver um subprograma para determinar o maior valor armazenado num vetor de n elementos
2. Desenvolver um algoritmo principal para testar o subprograma acima
3. Desenvolver um conjunto de subprogramas para determinar certos valores para os dados armazenados num vetor de números reais:
  1. A média
  2. A mediana
  3. O desvio padrão
  4. O número de valores acima da médiaObs. Esses subprogramas podem utilizar subprogramas definidos anteriormente, e também uns aos outros.
4. Desenvolver um algoritmo principal para testar os subprogramas desenvolvidos no item acima.

# Vetores em PASCAL

## Vetores

- Tipos Compostos
  - Conjunto de vários elementos de mesmo tipo
- Um único nome. Ex: V
- Um tipo. Ex: V:inteiro

v 

|         |         |         |         |         |         |         |
|---------|---------|---------|---------|---------|---------|---------|
| inteiro | inteiro | inteiro | inteiro | inteiro | inteiro | inteiro |
|---------|---------|---------|---------|---------|---------|---------|

- Armazenamento em sequência. V:inteiro[1..7]
- Um índice.
  - Acesso direto: V[i]

## Exemplos

tipo frase = caracter[0..80]

variável  
valor: real[1..10]  
linha: frase

valor[1] ← 0.35  
linha[0] ← ':'

## Declaração (PASCAL) de Variável Indexada Unidimensional

- Lembre-se que o espaço de armazenamento de um vetor é reservado no momento de sua declaração
- Deve ser especificado o menor e o maior índice do conjunto
- Deve ser especificado o tipo dos elementos do conjunto
- Exemplo:

```
var
X:array[1..200]of real;
```

## Declaração (PASCAL) de Variável Indexada Unidimensional

- Deve ser especificado o número máximo de elementos do conjunto
- Deve ser especificado o menor índice e o maior índice do conjunto
- Deve ser especificado o tipo dos elementos do conjunto

```
var
X:array[1..200]of real;
```

## Exemplo

algoritmo maior  
declarações

```
leia (n)
para j de 1 até n faça
  leia nro[j]
fim para
maior ← nro[1]
para i de 2 até n passo 1 faça
  se nro[i] > maior então
    maior ← nro[i]
  fim se
fim para
escreva (maior)
fim
```

## Exemplo

```
program vetor1;
var
  n,i,maior:integer;
  nro:array[1..100] of integer;
begin
  write('forneca a quantidade de numeros: '); readln(n);
  writeln;
  for i:=1 to n do
  begin
    write('forneca o ',i,'o. numero: ');
    readln(nro[i]);
  end;
  writeln;
  maior:=nro[1];
  for i:=2 to n do
  begin
    if (nro[i] > maior)then maior:=nro[i];
  end;
  writeln('maior numero: ',maior);
  readln;
end.
```

## Vetores em Subprogramas

- Em Pascal, para que um vetor possa ser usado como parâmetro, seu tipo tem que ser definido previamente.;
- Exemplo:

```
program EXEMPLO;
const
  tf = 100;
type
  tvetor = array [1..tf] of integer;
var
  Vet1: tvetor;
```

## Exemplo

```
constante
  max_el = 100
tipo
  vet_int = inteiro[1..max_el];

Subprograma maior_elemento (valores,n):inteiro
e: valores:vet_int
n: inteiro
r: o maior elemento do vetor, inteiro
declarações
início
  maior ← valores[1]
  para i de 2 até n passo 1 faça
    se valores[i] > maior então
      maior ← valores[i]
  fim se
fim para
retorne(maior)
fim
```

## Exemplo

```
program VETOR1;
const
  max_el = 100;
type
  vet_int = array[1..max_el] of integer;
var
  n,i:integer;
  nro:vet_int;
Function maior_elemento(valores:vet_int;n:integer):integer;
var
  maior,i:integer;
begin
  maior := valores[1];
  for i:=1 to n do
  begin
    if valores[i]>maior then maior := valores[i];
  end;
  maior_elemento := maior;
end;
```

## Exemplo

```
begin
  write('FORNECA A QUANTIDADE DE NUMEROS: '); readln(N);
  writeln;
  for i:=1 to n do
  begin
    write('FORNECA O ',I,'o. NUMERO: ');
    readln(nro[I]);
  end;
  writeln;
  writeln('MAIOR NUMERO: ', maior_elemento(nro,n));
  readln;
end.
```

## Inicialização de Vetores em Pascal

O Pascal admite inicialização de vetores constantes como no exemplo abaixo:

```
const
  nome : array [1..i2] of tipo_simples = (c1,c2,...,c_max)
ou
  nome: tipo = (c1,c2,...,c_max)
```

onde:

**i1..i2** é o intervalo de indexação do vetor,  
**tipo\_simples** é o tipo elementar do conteúdo do vetor,  
**tipo** é um tipo vetor definido pelo programador previamente e  
**c1,c2,...,c\_max** são constantes do tipo elementar armazenado no vetor.

Alguns exemplos:

```
type
  vet = array [1..6] of integer.
const
  v: vet = (4,5,4,3,3,1)
  c: array [1..3] of char = ('A', 'B', 'C')
```

## Exemplo

```
program vetores;
type
  vet = array[1..10] of integer;
const
  v1:vet = (0,1,2,3,4,5,6,7,8,9);
  c:array[1..3] of char = ('A','B','C');
var
  v:vet;
begin
  writeln(v1[1], v1[2], v1[3]);
  writeln(c[1],c[2],c[3]);
  readln;
end.
```

## Sugestão

- Variáveis compostas (vetores e matrizes) com um número razoável de elementos devem ser passadas em subprogramas sempre por referência, mesmo quando forem parâmetros apenas de entrada
- Esta prática deve ser acompanhada de comentário adequado, logo abaixo do cabeçalho do subprograma, indicando quais parâmetros são de entrada, quais são de saída, e quais são de entrada e saída