

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

Disciplina: 113475

Profa. Carla Denise Castanho

Universidade de Brasília - UnB
Instituto de Ciências Exatas - IE
Departamento de Ciência da Computação - CIC

6. VETORES



Conjuntos Homogêneos

- ▶ Imagine que você precisa fazer um programa para calcular as médias finais dos alunos de APC.
- ▶ Para cada aluno, você deve ler as notas de 3 provas. Deve então calcular e mostrar a média de cada um.

Conjuntos Homogêneos

- Com o que aprendemos até agora, é possível resolver esse problema. No entanto, você precisaria de:

40 x 3 provas = **120 variáveis!!**

Conjuntos Homogêneos

Exemplo - Variáveis!! Infinitas variáveis!!!

Algoritmo NotaCB

Variáveis

```
p1_a1, p1_a2, ..., p1_a40 : real  
p2_a1, p2_a2, ..., p2_a40 : real  
p3_a1, p3_a2, ..., p3_a40 : real
```

} **120 VARIÁVEIS!!!** 🤯

Início

```
Escreva ("Informe a nota da Prova 1 do Aluno 1:")  
Leia (p1_a1)  
...
```

} **120 Leituras!!!
40 Cálculos e Escritas!!!** 😞

Fim

Conjuntos Homogêneos

- ▶ E se fosse um programa para calcular as médias da UnB inteira?
- ▶ E se você precisar adicionar uma nota? Ou fazer qualquer alteração no cálculo?
- ▶ Certamente deve existir um jeito melhor de resolver esse problema...

Conjuntos Homogêneos

- ▶ Para trabalhar com muitos dados de **mesmo tipo**, nós usamos a noção de **conjuntos homogêneos**.
- ▶ Chamamos esse tipo de **estrutura de dados** de **vetores**.
- ▶ O vetor tem um **tamanho** definido (número de elementos) e cada elemento é referenciado por meio de um **índice**.

Vetores

- ▶ Por exemplo, para nos referirmos ao **i-ésimo** elemento, usamos **vetor[i]** - lê-se “vetor, índice *i*”.
- ▶ Em algumas linguagens, o primeiro elemento do vetor é **vetor[1]**, mas, na maioria das linguagens, por uma questão de eficiência, **começamos a contar do zero** e vamos até (tamanho do vetor - 1)!
- ▶ Na linguagem C, por exemplo, um vetor de **40** elementos vai de **vetor[0]** até **vetor[39]**...
- ▶ Vamos usar essa mesma convenção em pseudocódigo.

Vetores

- Para declarar um vetor em pseudocódigo, utilizamos a seguinte notação:

Sintaxe para declarar um Vetor

```
<nome> : vetor [<tamanho>] de <tipo>
```

Exemplos

Variáveis

```
primos : vetor [10] de inteiros  
notas : vetor [100] de reais  
sexo : vetor [30] de caracteres
```

- Lembre-se: todos os elementos do vetor têm o mesmo tipo!

Vetores - Exemplo

Exemplo - Cálculo de Notas de CB

Algoritmo NotaCB_Esperto

Variáveis

```
prova1 : vetor [40] de reais
prova2 : vetor [40] de reais
prova3 : vetor [40] de reais
media : vetor [40] de reais
i : inteiro
```

Início

```
Para i ← 0 até 39 faça
    Leia (prova1[i])
    Leia (prova2[i])
    Leia (prova3[i])
```

Fim-para

```
Para i ← 0 até 39 faça
    media[i] ← (prova1[i] + prova2[i] + prova3[i]) / 3
```

Fim-para

```
Para i ← 0 até 39 faça
    Escreva ("Média do Aluno ", i+1, ": ", media[i])
```

Fim-para

Fim

Vetores - Exemplo

Exemplo - Cálculo de Notas de CB

Algoritmo NotaCB_Esperto

Variáveis

```
prova1 : vetor [40] de reais
prova2 : vetor [40] de reais
prova3 : vetor [40] de reais
media : vetor [40] de reais
i : inteiro
```

Início

```
Para i ← 0 até 39 faça
    Leia (prova1[i])
    Leia (prova2[i])
    Leia (prova3[i])
```

ENTRADA DE DADOS

Fim-para

```
Para i ← 0 até 39 faça
```

```
    media[i] ← (prova1[i] + prova2[i] + prova3[i]) / 3
```

PROCESSAMENTO

Fim-para

```
Para i ← 0 até 39 faça
```

```
    Escreva ("Média do Aluno ", i+1, ": ", media[i])
```

SAÍDA DE DADOS

Fim-para

Fim

Vetores na Linguagem C

- Para declarar um vetor em C, utilizamos a seguinte notação:

Sintaxe para declarar um Vetor em C

```
<tipo> <nome> [<tamanho>];
```

Exemplos

```
int primos[10];  
float notas[100];  
char sexo[30];
```

Vetores na Linguagem C

Exemplo - Cálculo de Notas de CB em C

```
#include <stdio.h>

int main () {
    float prova1[40], prova2[40], prova3[40], media[40];
    int i;

    for (i = 0; i < 40; i++) {
        printf("Informe as notas do aluno %d:\n", i);
        scanf("%f%f%f", &prova1[i], &prova2[i], &prova3[i]);
    }
    for (i = 0; i < 40; i++) {
        media[i] = ((prova1[i] + prova2[i] + prova3[i]) / 3);
    }
    for (i = 0; i < 40; i++) {
        printf("Media do aluno %d: %.2f.\n", i+1, media[i]);
    }

    return 0;
}
```

Vetores - Cuidado com o índice!

- ▶ Quando utilizamos um vetor **N**, devemos sempre garantir que o índice está entre **0** e **N-1**!
- ▶ Se você tentar utilizar um índice negativo ou maior ou igual a N, seu programa se comportará de maneira inesperada. Geralmente acontece um *segmentation fault*, isto é, seu programa tenta acessar uma área de memória inválida.
- ▶ Esse tipo de erro não é detectado em tempo de compilação, apenas quando você executa o programa!

Número Variável de Elementos

- ▶ Você não precisa usar todos os N elementos do vetor!
- ▶ É comum declararmos um vetor **maior que o necessário** e perguntarmos ao usuário quantas posições ele vai precisar.
- ▶ Nesse caso, nosso índice deve variar de **0 até esse novo limite**, obviamente menor ou igual ao tamanho declarado do vetor.
- ▶ Veja o exemplo no próximo slide...

Número Variável de Elementos

Exemplo - Cálculo de Notas de CB

Algoritmo NotaCB_Variável

Variáveis

```
prova1 : vetor [100] de reais
prova2 : vetor [100] de reais
prova3 : vetor [100] de reais
media : vetor [100] de reais
i, nro_alunos : inteiro
```

Início

```
Escreva ("Informe o número de alunos:")
Leia (nro_alunos)
Para i ← 0 até nro_alunos - 1 faça
    Leia (prova1[i])
    Leia (prova2[i])
    Leia (prova3[i])
Fim-para
Para i ← 0 até nro_alunos - 1 faça
    media[i] ← ((prova1[i] + prova2[i] + prova3[i]) / 3)
Fim-para
Para i ← 0 até nro_alunos - 1 faça
    Escreva ("Média do Aluno ", i+1, ": ", media[i])
Fim-para
Fim
```


Número Variável de Elementos

Exemplo - Cálculo de Notas de CB

Algoritmo NotaCB_Variável

Variáveis

```
prova1 : vetor [100] de reais
prova2 : vetor [100] de reais
prova3 : vetor [100] de reais
media : vetor [100] de reais
i, nro_alunos : inteiro
```

Início

```
Escreva ("Informe o número de alunos:")
Leia (nro_alunos)
```

```
Para i ← 0 até nro_alunos - 1 faça
    Leia (prova1[i])
    Leia (prova2[i])
    Leia (prova3[i])
```

Fim-para

```
Para i ← 0 até nro_alunos - 1 faça
    media[i] ← ((prova1[i] + prova2[i] + prova3[i]) / 3)
```

Fim-para

```
Para i ← 0 até nro_alunos - 1 faça
    Escreva ("Média do Aluno ", i+1, ": ", media[i])
```

Fim-para

Fim

Número Variável de Elementos

Exemplo - Cálculo de Notas de CB

Algoritmo NotaCB_Variável

Variáveis

```
prova1 : vetor [100] de reais
prova2 : vetor [100] de reais
prova3 : vetor [100] de reais
media : vetor [100] de reais
i, nro_alunos : inteiro
```

Início

```
Escreva ("Informe o número de alunos:")
```

```
Leia (nro_alunos)
```

```
Para i ← 0 até nro_alunos - 1 faça
```

```
    Leia (prova1[i])
```

```
    Leia (prova2[i])
```

```
    Leia (prova3[i])
```

```
Fim-para
```

```
Para i ← 0 até nro_alunos - 1 faça
```

```
    media[i] ← ((prova1[i] + prova2[i] + prova3[i]) / 3)
```

```
Fim-para
```

```
Para i ← 0 até nro_alunos - 1 faça
```

```
    Escreva ("Média do Aluno ", i+1, ": ", media[i])
```

```
Fim-para
```

Fim

Número Variável de Elementos

Programa C do Algoritmo Anterior

```
#include <stdio.h>

int main () {
    float prova1[100], prova2[100], prova3[100], media[100];
    int i, nro_alunos;

    printf("Informe o total de alunos:\n");
    scanf("%d", &nro_alunos);

    for (i = 0; (i < nro_alunos) && (i < 100); i++) {
        printf("Informe as notas do aluno %d:\n", i);
        scanf("%f%f%f", &prova1[i], &prova2[i], &prova3[i]);
    }
    for (i = 0; (i < nro_alunos) && (i < 100); i++) {
        media[i] = ((prova1[i] + prova2[i] + prova3[i]) / 3);
    }
    for (i = 0; (i < nro_alunos) && (i < 100); i++) {
        printf("Media do aluno %d: %.2f.\n", i+1, media[i]);
    }

    return 0;
}
```

Número Variável de Elementos

Programa C do Algoritmo Anterior

```
#include <stdio.h>
```

TEMOS QUE FAZER DOIS TESTES!!

```
int main () {  
    float prova1[100], prova2[100], prova3[100], media[100];  
    int i, nro_alunos;
```

```
    printf("Informe o total de alunos:\n");  
    scanf("%d", &nro_alunos);
```

Garante que apenas o número informado de alunos será processado!

```
    for (i = 0; (i < nro_alunos) && (i < 100); i++) {  
        printf("Informe as notas do aluno %d:\n", i+1);  
        scanf("%f%f%f", &prova1[i], &prova2[i], &prova3[i]);  
    }
```

Garante que o índice não ultrapassa o tamanho do vetor!

```
    for (i = 0; (i < nro_alunos) && (i < 100); i++) {  
        media[i] = (prova1[i] + prova2[i] + prova3[i]) / 3;  
    }  
    for (i = 0; (i < nro_alunos) && (i < 100); i++) {  
        printf("Media do aluno %d: %.2f.\n", i, media[i]);  
    }
```

```
    return 0;
```

```
}
```

Vetores - Exemplos

- Ex. 1: Escreva um algoritmo que leia N notas (N será informado pelo usuário e será sempre menor ou igual a 100), calcule a média das notas e imprima na tela todas as notas que ficaram abaixo dessa média.

Vetores - Exemplos

Resolução do Exemplo Anterior

Algoritmo Notas

Variáveis

i, n : inteiro
media : real
notas : vetor [100] de reais

Início

Escreva ("Digite o número de alunos:")

Leia (n)

media ← 0

Para i ← 0 **até** n - 1 **faça**

Escreva ("Digite a nota do aluno ", i + 1)

Leia (notas[i])

 media ← media + notas[i]

Fim-Para

media ← media / n

Escreva ("A média das notas informadas é ", media)

Para i ← 0 **até** n - 1 **faça**

Se (notas[i] < media) **então**

Escreva ("A nota do aluno ", i+1, ", ", notas[i], ", está abaixo da média.")

Fim-Se

Fim-Para

Fim

Vetores - Exemplos

Programa C do Algoritmo Anterior

```
#include <stdio.h>

int main () {
    float media, notas[100];
    int i, n;

    printf("Informe o total de alunos:\n");
    scanf("%d", &n);
    media = 0;
    for (i = 0; (i < n) && (i < 100); i++) {
        printf("Informe a nota do aluno %d:\n", i+1);
        scanf("%f", &notas[i]);
        media += notas[i];
    }
    media /= n; /* media = (media / n) */
    printf("A media das notas informadas eh %.2f.\n", media);
    for (i = 0; (i < n) && (i < 100); i++) {
        if (notas[i] < media)
            printf("A nota do aluno %d, %.2f, esta abaixo da media.\n", i+1, notas[i]);
    }

    return 0;
}
```

Vetores - Exemplos

- Ex. 2: Faça um algoritmo que leia um conjunto de N inteiros (N será lido e é sempre menor ou igual a 100), encontre e mostre o maior deles.

Vetores - Exemplos

Resolução do Exemplo Anterior

Algoritmo Números

Variáveis

i, n, maior : inteiro
numeros : vetor [100] de inteiros

Início

Escreva ("Informe a quantidade de números a serem lidos:")

Leia (n)

Para i ← 0 até n - 1 faça

Escreva ("Digite o ", i + 1, "o numero:")

Leia (numeros[i])

Fim-Para

maior ← numeros[0]

Para i ← 1 até n - 1 faça

Se (numeros[i] > maior) então

maior ← numeros[i]

Fim-Se

Fim-Para

Escreva ("O maior número é ", maior, ".")

Fim

Vetores - Exemplos

Programa C do Algoritmo Anterior

```
#include <stdio.h>

int main () {
    int i, n, maior, numeros[100];

    printf("Informe a quantidade de numeros a serem lidos:\n");
    scanf("%d", &n);
    for (i = 0; (i < n) && (i < 100); i++) {
        printf("Digite o %do numero:\n", i + 1);
        scanf("%d", &numeros[i]);
    }
    maior = numeros[0];
    for (i = 1; (i < n) && (i < 100); i++) {
        if (numeros[i] > maior )
            maior = numeros[i];
    }
    printf("O maior numero eh %d.\n", maior);

    return 0;
}
```

Vetores como Parâmetros de Funções

- ▶ Na linguagem C (e no nosso pseudocódigo), **vetores** e **ponteiros** são **intercambiáveis**. O nome de um vetor é também um ponteiro para o **primeiro elemento** do vetor!
- ▶ Portanto, para passar um vetor como parâmetro de uma função, você tem duas opções:

Sintaxe - Vetores como Parâmetros de Funções em Pseudocódigo

```
Função RecebeVetor(<nome> : *<tipo> )  
Função RecebeVetor(<nome> : vetor de <tipo> )
```

Sintaxe - Vetores como Parâmetros de Funções em C

```
void recebe_vetor(<tipo> *<nome>) /* um ponteiro */  
void recebe_vetor(<tipo> <nome>[]) /* um vetor */
```

- ▶ Observe que **um vetor é sempre passado por referência!**
- ▶ Idealmente, devemos sempre passar também o **tamanho** do vetor, a função não tem como saber essa informação, sabe apenas que se trata de um vetor!

Vetores - Exemplos

- Ex. 3: Escreva um algoritmo que leia um tamanho N , em seguida, para dois vetores $V1$ e $V2$, leia e armazene N números inteiros. Este algoritmo deve utilizar uma função que receba vetores de inteiros, bem como o tamanho dos vetores, e retorne 1 se os vetores forem iguais e 0 caso contrário. O algoritmo deve mostrar uma mensagem na tela indicando se os vetores são iguais ou não.

Vetores - Exemplos

Resolução do Exemplo Anterior

Algoritmo ComparaVetores

Função Compara(a : vetor de inteiros, b : vetor de inteiros, tam : inteiro) : inteiro

Variáveis

i : inteiro

Início

Para i ← 0 até tam - 1 faça

Se a[i] ≠ b[i] então

retorne 0

Fim-Para

retorne 1

Fim

Variáveis

n, i : inteiro

v1, v2 : vetor [100] de inteiros

Início

Leia (n)

Para i ← 0 até n - 1 faça

Leia (v1[i])

Para i ← 0 até n - 1 faça

Leia (v2[i])

Se Compara(v1, v2, n) = 1 então

Escreva ("Os vetores são iguais.")

Senão

Escreva ("Os vetores são diferentes.")

Fim

Vetores - Exemplos

Programa C do Algoritmo Anterior

```
#include <stdio.h>

int compara (int a[], int b[], int tam) {
    int i;
    for (i = 0; i < tam; i++) {
        if (a[i] != b[i])
            return 0;
    }
    return 1;
}

int main () {
    int v1[100], v2[100], n;

    printf("Informe o tamanho dos vetores:\n");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
        scanf("%d", &v1[i]);
    for (i = 0; i < n; i++)
        scanf("%d", &v2[i]);

    if (compara(v1, v2, n) == 1)
        printf("Os vetores são iguais.\n");
    else
        printf("Os vetores são diferentes.\n");

    return 0;
}
```

Vetores - Exemplos

Programa C do Algoritmo Anterior

```
#include <stdio.h>

int compara (int a[], int b[], int tam) {
    int i;
    for (i = 0; i < tam; i++) {
        if (a[i] != b[i])
            return 0;
    }
    return 1;
}

int main () {
    int v1[100], v2[100], n;

    printf("Informe o tamanho dos vetores:\n");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
        scanf("%d", &v1[i]);
    for (i = 0; i < n; i++)
        scanf("%d", &v2[i]);

    if (compara(v1, v2, n) == 1)
        printf("Os vetores são iguais.\n");
    else
        printf("Os vetores são diferentes.\n");

    return 0;
}
```

Poderíamos declarar também:

```
int compara(int *a, int *b, int tam)
```

E a função continuaria igual! Vetores e ponteiros são equivalentes!

A chamada da função também ficaria igual! Lembre-se: vetores já são ponteiros, por isso não precisa passar o endereço!

É por isso que, quando você vai ler uma string com scanf, não precisa de &.

Vetores - Exercícios

1. Escreva um algoritmo que leia dois vetores de 25 inteiros cada (um vetor e depois o outro), crie um vetor de 50 inteiros que seja o resultado da intercalação dos elementos dos outros dois vetores. Ao final, mostre o conteúdo do novo vetor.
2. Escreva um algoritmo que leia um vetor de inteiros com 20 posições e troque todos os elementos do vetor que foram negativos por zero. O algoritmo deve mostrar o vetor modificado. Devem ser criadas funções para:
 1. ler o vetor;
 2. trocar os elementos negativos por zero;
 3. mostrar o vetor na tela.No corpo do programa principal devem ser chamadas as três funções acima. O vetor deve ser passado como parâmetro adequadamente.
3. Crie um algoritmo com uma função Busca que recebe dois vetores COD e PRECO, de 10 posições, bem como um código inteiro, e retorna o respectivo preço para o código informado. Crie também uma função para ler os vetores. No programa principal, você deve utilizar as funções para ler uma lista de 10 produtos com código e preço, bem como o código de um produto a ser pesquisado, em seguida, buscar e mostrar o preço do produto cujo código foi informado.