Aula – Programação II (INF 09330)

UFES
Informática

Variáveis Indexadas

Prof. Thiago Oliveira dos Santos
Departamento de Informática
Universidade Federal do Espírito Santo

2015

Visão Geral da Aula

UFES
Informática

- Introdução
- Variáveis indexadas unidimensionais (Vetores)
- Strings
- Variáveis indexadas bidimensionais (Matrizes)

Introdução

UFES
Informática

Variáveis

- Variáveis são abstrações que facilitam acesso à memória
- São representadas por um símbolo

Limitações das Variáveis Comuns

- Não permitem armazenar vários valores de mesmo tipo
- Exemplo de problemas desse tipo
  - Dado a idade de um grupo de pessoas, contar numero de indivíduos acima da média
  - Esse tipo de problema requer solução complexa sem uso de recursos adicionais

Introdução

UFES
Informática

Variáveis Indexadas

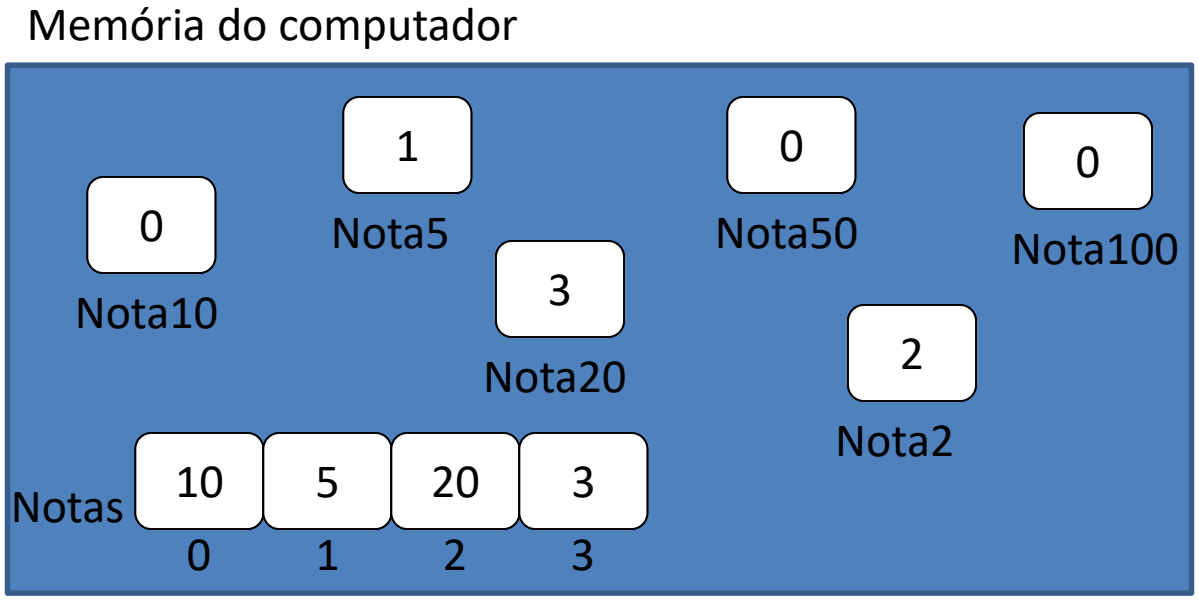
- Possuem um mesmo nome
- São variáveis que são diferenciadas através de um índice
- Podem possuir mais de uma dimensão
  - Variáveis indexadas unidimensionais (Vetores)
  - Variáveis indexadas bidimensionais (Matriz)
  - Variáveis indexadas multidimensionais

Introdução

UFES
Informática

Variáveis Indexadas vs Variáveis Comuns

- Podem ser acessadas na seqüência



Variáveis Indexadas Unidimensionais (Vetores)

UFES
Informática

Nome

- Segue a mesma regra das variáveis comuns

Declaração

- Deve ser feita antes do uso (geralmente no início do bloco)
- Sintaxe
  - <tipo> <nome>[<tamanho>];
    - <tipo> = tipo de dados válido
    - <nome> = nomes da variável
    - <tamanho> = tamanho do vetor
- Exemplo
  - float velocidades[10]; int idades[30], char nome[10];

Variáveis Indexadas Unidimensionais (Vetores)

UFES
Informática

Acesso aos Dados

- Atribuição entre vetores não funciona ← Antenção!
  - Deve ser feito item a item
- Utiliza-se o operador [] para acesso aos itens
- Índices dos itens vão de *0* a *n-1*
- Exemplo 1
  - int idadeTemp = idades[0];
  - float velocidadeTemp = velocidades[9];

Variáveis Indexadas Unidimensionais (Vetores)

UFES
Informática

Inicialização

- Pode ser feita em lote com o operador {}
  - Exemplo
    - float velocidades[3] = { 100.0, 80.0, 50.0 };
    - int num[4] = { 1, 2, 3, 4 };

Variáveis Indexadas Unidimensionais (Vetores)

UFES
Informática

Acesso aos Dados

- Exemplo

```
int main()
{
   int n[3];
   printf("Digite um num\n");
   scanf("%d", &n[0]);
   printf("Digite um num\n");
   scanf("%d", &n[1]);
   printf("Digite um num\n");
   scanf("%d", &n[2]);

   printf("num1 = %d\n", n[0]);
   printf("num2 = %d\n", n[1]);
   printf("num3 = %d\n", n[2]);
   return 0;
}
```

Variáveis Indexadas Unidimensionais (Vetores)

UFES
Informática

Acesso Iterativo

- Possibilita percorrer uma seqüência de variáveis
- Exemplo

```
#define TAM 10
int main()
{
   int i, numeros[TAM];
   printf("Informe 10 numeros inteiros:\n");
   for (i = 0; i < TAM; i++)
      scanf("%d", &numeros[i]);

   printf("Ordem inversa:\n");
   for (i = TAM-1; i >= 0; i--) {
      printf("%d\n", numeros[i]);
   }
   return 0;
}
```

Variáveis Indexadas Unidimensionais (Vetores)

UFES
Informática

Tamanho Dinâmico

- O tamanho é definido durante a execução
- Exemplo

```
int main()
{
   int i, tam;
   printf("Informe a quantidade de numeros:\n");
   scanf("%d", &tam);
   int numeros[tam];
   printf("Informe %d numeros inteiros:\n", tam);
   for (i = 0; i < tam; i++)
      scanf("%d", &numeros[i]);
   printf("Ordem inversa:\n");
   for (i = tam-1; i >= 0; i--) {
      printf("%d\n", numeros[i]);
   }
   return 0;
}
```

Variáveis Indexadas Unidimensionais (Vetores)

UFES
Informática

Uso de Vetores

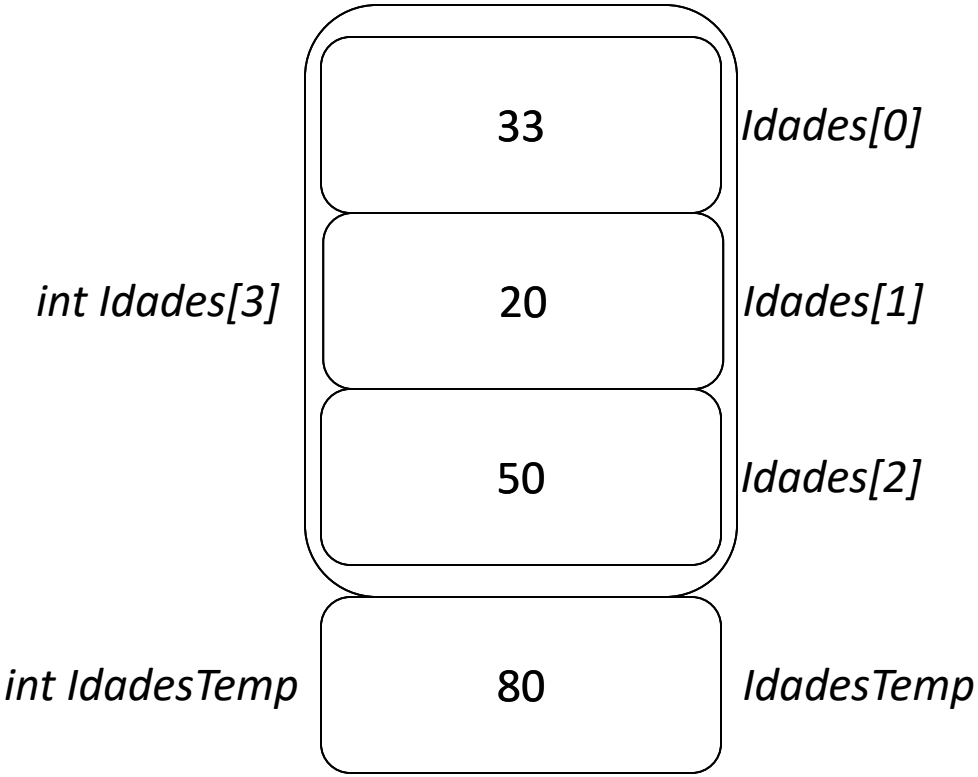
- Impressão dos alunos acima da média da turma
- Exemplo

```
int main()
{
   int i, tam;    float soma = 0, media;
   printf("Informe a quantidade de numeros:\n");
   scanf("%d", &tam);
   int numeros[tam];
   printf("Informe %d numeros inteiros:\n", tam);
   for (i = 0; i < tam; i++){
      scanf("%d", &numeros[i]);
      soma = soma + numeros[i];
   }
   media = soma / (float)tam;
   printf("Alunos acima da media, %f:\n", media);
   for (i = 0; i < tam; i++) {
      if (numeros[i] > media){
         printf("i %d => nota %d\n", i, numeros[i]);
      }
   }
   return 0;
}
```

Variáveis Indexadas Unidimensionais (Vetores)

UFES
Informática

Organização em Memória



Vetores como Argumento de Função

UFES
Informática

Unidimensional

- Vetores podem ser passados como argumento para função
- Existem 3 formas
- Exemplo para um vetor *int vetor[10];*
  - void func(int* vet);
  - void func(int vet[]);
  - void func(int vet[10]); CUIDADO! VETOR NÃO É ALOCADO

Vetores como Argumento de Função

UFES
Informática

Unidimensional

- Exemplo 1

```
int main()
{
   int tam;
   printf("Informe a quantidade de numeros:\n");
   scanf("%d", &tam);
   int numeros[tam];
   PreencheVetor(numeros, tam);
   ImprimeOrdemInversa(numeros, tam);
   return 0;
}
```

Vetores como Argumento de Função

UFES
Informática

Unidimensional

- Exemplo 1

```
void PreencheVetor(int v[], int tam){
   int i;
   printf("Informe %d numeros inteiros:\n", tam);
   for (i = 0; i < tam; i++)
      scanf("%d", &v[i]);
}
void ImprimeOrdemInversa(int v[], int tam){
   int i;
   printf("Ordem inversa:\n");
   for (i = tam-1; i >= 0; i--) {
      printf("%d\n", v[i]);
   }
}
```

Vetores como Argumento de Função

UFES
Informática

Unidimensional

- Exemplo 2
  - Ordenar um vetor de inteiros

Vetores como Argumento de Função

UFES
Informática

Unidimensional

- Exemplo 2
  - Ordenar um vetor de inteiros

```
void OrdeneCrescente(int vet[], int qtd){
    int i, idxMenor, aux;

    for(i = 0; i < qtd-1; i++)
    {
      idxMenor = AcharMenorEntreAeB(vet, i+1, qtd-1);
      if (EhMenorAqB(vet[idxMenor], vet[i]) ){
        Trocavalor (vet, idxMenor, i);
      }
    }

}
```

Vetores como Argumento de Função

UFES
Informática

Unidimensional

- Exemplo 2
  - Ordenar um vetor de inteiros

```
int AcharMenorEntreAeB(int vet[], int a, int b){
    int i, idx, menor;
    menor = vet[a];
    idx = a;
    for(i = a+1; i <= b; i++)
    {
        if ( EhMenorAqB(vet[i], menor)){
            menor = vet[i];
            idx = i;
        }
    }
    return idx;
}
```

Vetores como Argumento de Função

UFES
Informática

Unidimensional

- Exemplo 2
  - Ordenar um vetor de inteiros

```
int EhMenorAqB(int a, int b){
   return a < b;
}


void Trocavalor(int vet[], int i, int j){
   int aux;
   aux = vet[j];
   vet[j] = vet[i];
   vet[i] = aux;
}
```

Vetores de *Structs*

UFES
Informática

Unidimensional

- Vetores podem ser de qualquer tipo
  - Inclusive de um tipo definido pelo usuário
- Utilizando o exemplo do tipo tData
  - Imagine que se queira ordenar uma série de datas de aniversários

```
#define QTD 5
#include “tData.h”
int main()
{
   tData datas[QTD];
   datas[0] = InicializaDataParam(13, 02, 2015);
   datas[1] = InicializaDataParam(13, 02, 2014);
   datas[2] = InicializaDataParam(13, 05, 2015);
   datas[3] = InicializaDataParam(13, 10, 2012);
   datas[4] = InicializaDataParam(11, 02, 2015);
   ApresentaDatas(datas, QTD);
   OrdeneDatasCrescente(datas, QTD);
   ApresentaDatas(datas, QTD);
   return 0;
}
```

```
tData InicializaDataParam( int a_dia, int a_mes, int a_ano)
{
   tData data;
   data.dia = a_dia;
   data.mes = a_mes;
   data.ano = a_ano;

   return data;
}
```

Vetores de *Structs*

UFES
Informática

Unidimensional

```
void OrdeneDatasCrescente(tData vet[], int qtd){
    int i, idxMenor;

    for(i = 0; i < qtd-1; i++)
    {
      idxMenor = AcharMenorEntreAeB(vet, i+1, qtd-1);
      if ( EhMenorDataAqDataB(vet[idxMenor], vet[i]) ){
          Trocavalor (vet, idxMenor, i);
      }
    }

}
```

Vetores de *Structs*

UFES
Informática

Unidimensional

```
int AcharMenorEntreAeB(tData vet[], int a, int b){
    int i, idx;
    tData menor;

    menor = vet[a];
    idx = a;
    for(i = a+1; i <= b; i++){
        if ( EhMenorDataAqDataB(vet[i], menor) ){
            menor = vet[i];
            idx = i;
        }
    }
    return idx;
}
```

Vetores de *Structs*

UFES
Informática

Unidimensional

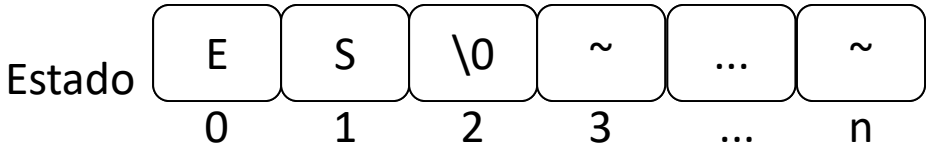
```
int EhMenorDataAqDataB(tData a, tData b){
   if (a.ano < b.ano)
      return 1;
   else if (a.ano == b.ano && a.mes < b.mes)
      return 1;
   else if (a.ano == b.ano && a.mes == b.mes && a.dia < b.dia)
      return 1;

   return 0;
}

void Trocavalor(tData vet[], int i, int j){
   tData aux;
   aux = vet[j];
   vet[j] = vet[i];
   vet[i] = aux;
}
```

Strings

UFES
Informática

Cadeia de Caracteres

- Geralmente utilizados na entrada e saída de dados
- Pode ser considerado um novo tipo
  - Estrutura de dados: vetores do tipo *char*
  - Toda string é terminada com um caractere especial '\0'
  - Deve ser utilizado usando operações próprias
- Assim como os vetores
  - Atribuição direta não é permitido
  - Deve-se copiar item a item
- Toda string é um vetor de char
  - Mas nem todo vetor de char é uma string em C



Strings

UFES
Informática

Inicialização

- Pode ser feita em lote com o operadores {} ou “”
  - Exemplo
    - char umNome[10] = { 'J', 'o', 'a', 'o', '\0' };
    - char outroNome[10] = "Joao"; ← Já inclui o \0, mas tem que haver espaço

Entrada

- Pode ser feita com %s no comando *scanf*
  - Exemplo: scanf(“%s”, umNome); //Note a retirada do &

Saída

Cuidado! Acesso inseguro.

- Pode ser feita com %s no comando *printf*
  - Exemplo: printf(“o nome eh: %s”, umNome);

Strings

UFES
Informática

Entrada

- Ler uma linha com *scanf* para um vetor de *char s*
  - scanf("%[^\n]", s);
    - Lê tudo até o \n, mas não lê o \n
    - Lembrar de consumir o \n se quiser ler outra linha
    - Lembrar que esse acesso não garante que o dado lido caberá em *s*
    - Para garantir, colocar um número máximo de caracteres a serem lidos entre o % e o [
      - Ex. para ler no máximo 100 caracteres: scanf("%100[^\n]", s);
  - scanf("%[a-z]", s);
    - Lê tudo enquanto for letra minúscula e para quando não for

String

UFES
Informática

Manipulação

- Exemplo

```
int main()
{
   int i = 0;
   char nome[100] = "Thiago Oliveira dos Santos\n";

   printf("Antes: %s", nome);
   while (nome[i] != '\0'){
      if (nome[i] == ' ')
         nome[i] = '_';
      i++;
   }
   printf("Depois: %s", nome);

   return 0;
}
```

Vetores em *Structs*

UFES
Informática

Unidimensional

- Se comportam ligeiramente diferente de vetores puros
- Assumem comportamento da struct
- Atribuição de structs contendo vetores é permitida
  - Copia todo o conteúdo
- Exemplo

```
typedef struct{
   char nome[100];
   int idade;
} tPessoa;
```

```
int main()
{
   tPessoa eu = {"Thiago Oliveira dos Santos", 35}, voce = {"Voce", 25};

   printf("eu antes: %s, %d\n", eu.nome, eu.idade);
   printf("eu antes: %s, %d\n", voce.nome, voce.idade);
   eu = voce;
   printf("eu depois: %s, %d\n", eu.nome, eu.idade);
   printf("eu depois: %s, %d\n", voce.nome, voce.idade);
   return 0;
}
```

Variáveis Indexadas Bidimensionais (Matriz)

UFES
Informática

Declaração

- Sintaxe
  - <tipo> <nome>[<tamanho_n>] [<tamanho_m>];
    - <tipo> = tipo de dados válido
    - <nome> = nomes da variável
    - <tamanho_n> = número de linhas
    - <tamanho_m> = número de colunas

- Exemplo
  - float matriz[10][10]; int imagem[30][30];

Variáveis Indexadas Bidimensionais (Matriz)

UFES
Informática

Acesso aos Dados

- Utiliza-se o operador [] duas vezes
- Índices vão de *0* a *n-1*
- Exemplo
  - float idadeTemp = matriz[0][0];
  - int pixel = imagem[29][29];

Variáveis Indexadas Bidimensionais (Matriz)

UFES
Informática

Acesso Iterativo

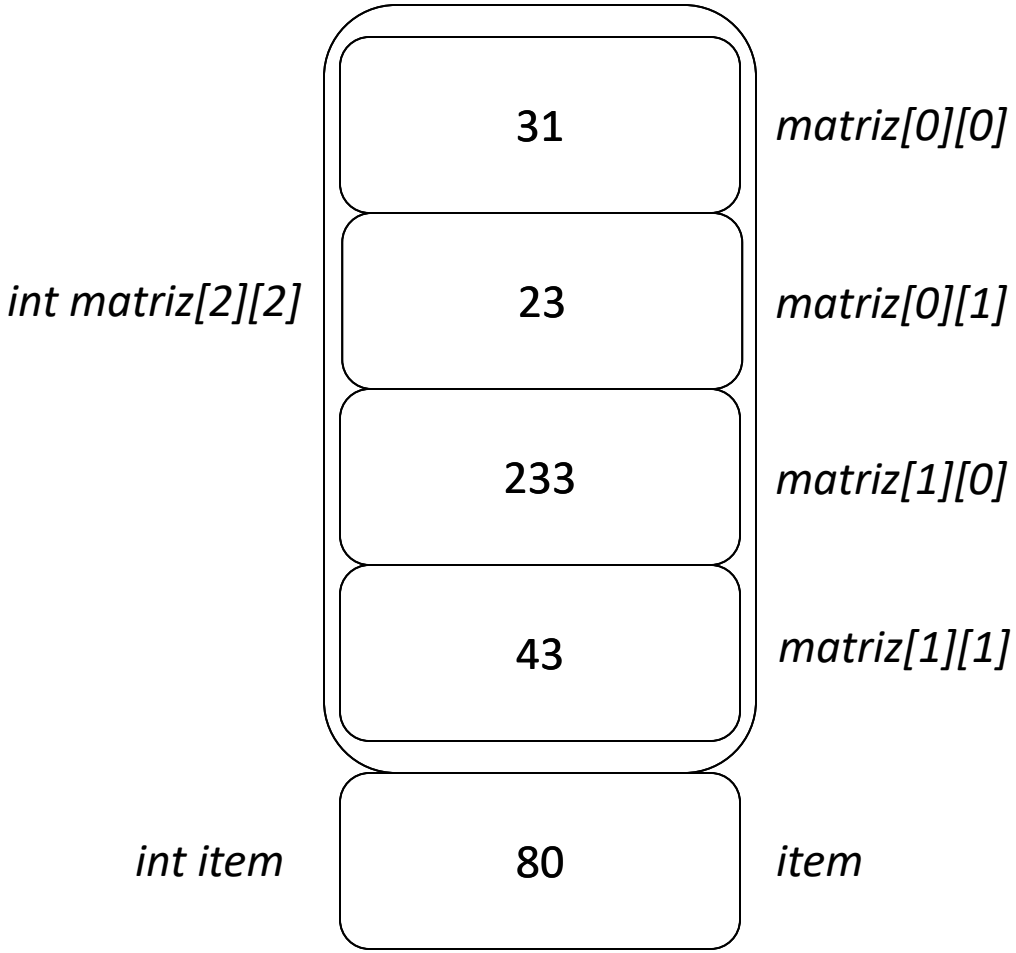
- Exemplo

```
int i, j, matriz[2][2];
printf("Informe 4 números inteiros:\n");
for (i = 0; i < 2; i++)
    for (j = 0; j < 2; j++)
        scanf("%d", &matriz[i][j]);

printf("Matriz:\n");
for (i = 0; i < 2; i++) {
    printf("\t");
    for (j = 0; j < 2; j++) {
        printf("%d ", matriz[i][j]);
    }
    printf("\n");
}
```

Variáveis Indexadas Unidimensionais (Vetores)

UFES
Informática

Organização em Memória (Row-Major)



	0	1
0	31	23
1	233	43

Variáveis Indexadas Bidimensionais (Matriz)

UFES
Informática

Inicialização

- Exemplo
  - int matriz[2][2] = { 1, 2, 3, 4 };
  - char vetorStrings[3][10] = { "Joao", "Maria", "Jose" };

Variáveis Indexadas Multidimensionais

UFES
Informática

Características

- Equivalente a bidimensional
- Inclui [] adicionais na declaração e no acesso a dados
- Exemplo
  - float matriz3D[10][10][10]; int multiDim[30][20][10][10];

Multidimensional como Argumento de Função

UFES
Informática

Bidimensional e Multidimensional

- Existe duas formas
- Exemplo para um vetor *int vetor[10][10];*
  - void func(int vet[10][10]);
  - void func(int vet[][10]);
- Exemplo para um vetor *int vetor[10][10][20];*
  - void func(int vet[10][10][20]);
  - void func(int vet[][10][20]);

Perguntas???

UFES
Informática

- Fazer exercícios da lista 5