

Linguagem de Programação I

Vetores, Matrizes e Strings

Dr^a Alana Moraes

Iesp Faculdades

Roteiro da Aula

- Vetores
- Strings
- Matrizes



Problema Inicial

- Preciso armazenar três notas para calcular a média do aluno.
- Como faríamos isso?



Problema Inicial

- Preciso armazenar três matrículas de alunos para cadastrar no meu curso.
- Como faríamos isso?

```
float matricula1, matricula2, matricula3;  
printf("Matrícula do aluno 1: ");  
scanf("%d", &matricula1);  
printf("Matrícula do aluno 2: ");  
scanf("%d", &matricula2);  
printf("Matrícula do aluno 3: ");  
scanf("%d", &matricula3);
```

Problema Inicial

- E se eu precisa criar um arquivo e armazenar 100 notas?
- Declaro 100 variáveis??
 - float matricula1, matricula2, matricula3, ..., matricula100;



Como resolveremos isto?

VETORES

- Vetor é uma coleção de variáveis do mesmo tipo que são referenciadas pelo mesmo nome.
- Um vetor consiste em locações contínuas de memória.
- As operações principais: declarar e inicializar, acessar, inserir e remover elementos.

Declaração de Vetores

tipo nome_var[tamanho];

onde:

- **tipo** é o tipo base do vetor e
- **tamanho** é a quantidade de elementos que o vetor conterá.

Exemplo:

float lista[10];

Declaração de Vetores

- É possível ainda já passar os valores diretamente para o vetor:
 - **int z[10] = {1,3,4,56,6,6,6,6,6,6};**

Vetores

Inserir

- Os vetores são acessados por meio de índices colocados entre colchetes.
- O índice do primeiro elemento do vetor é 0 (ZERO).
- Exemplo
int amostra[10]; /* vetor de 10 inteiros */
amostra[0] = 2; /* primeiro elemento */
amostra[9] = 7; /* último elemento

Vetores

Acessar

- Acessa os valores por meio dos índices numéricos.
- Manipula como qualquer variável.
- Exemplo

```
int teste[3] = {1,2,3}; //vetor de 3 inteiros  
printf("%d", teste[0]); //imprimir na tela  
int dobro = 2 * teste[2]; //op. de  
multiplicação
```

Vetores

Acessar

- Por meio de laços de repetição.

- Acessar a lista por completo:

```
int i;
```

```
for (i=0; i< tamanhoVetor; i++){
```

```
    printf("%dº elemento: %d \n", (i+1), lista[i]);
```

```
}
```

Remover Elementos

- Problema:
 - lista : 10 20 30 40 50
 - Apaguei: 30
 - Vetor atual : 10 20 40 50.

Remover Elementos

- Planejar
- Exemplo em código (vetor1.c)



Limite dos Vetores

- **C não faz checagem dos limites dos vetores**, isto é responsabilidade do programador. Logo, **o código a seguir não causará nenhum erro.**

```
int elementos[10];  
elementos[12] = 0;  
elementos[10] = 0;
```

Exercício

- Faça um programa que armazene uma lista de depósitos e saques de um cliente do banco Alana Corporation.
- Ao final imprima o saldo final deste cliente.



Strings

- Uma string é por definição, um vetor de caracteres terminado em \0.
- Então, para declarar a string, devemos declarar sempre um elemento a mais para o terminador.

String

- Exemplo:
 - `char mensagem[] = "Exemplo";`
 - Ficar  armazenado na mem ria como:

E x e m p l o \0

Strings

- Existem algumas funções bem úteis na biblioteca **string.h** (`#include <string.h>`)
- Algumas funções são importantes para se trabalhar com strings:

<code>strcpy</code>	Copia uma string em outra.
<code>strcmp</code>	Compara a cadeia apontada por <code>str1</code> à cadeia apontada por <code>str2</code>
<code>strlen</code>	Retorna o tamanho de uma string.
<code>strncat</code>	Concatena <code>n</code> caracteres da <code>string2</code> na <code>string1</code>
<code>strcasecmp</code>	Versão case insensitive de <code>strcmp()</code> .
<code>strchr</code>	Encontra a primeira ocorrência do caracter <code>c</code> na string.
<code>strrchr</code>	Encontra a última ocorrência do caracter <code>c</code> na string.
<code>strtok</code>	Quebra a string <code>s1</code> numa sequência de tokens delimitados por um ou mais caracteres de <code>s2</code> .

Exercício

- Faça um programa que leia uma string e retorne ela invertida.



Matriz

Exemplo de matrix

| 10 12 |

| 4 2 |

- Vetor de várias dimensões (exemplo vetor bidimensional, x e y)
- Declaração
 - `int matrizNome[2][2];`

Matriz

- Índices numéricos
 - Mesma forma de varrer de um vetor (com repetições aninhadas).
- Atribuição de valores diretamente:
 - `int matrizNome[2][2] = {{10,12},{4,2}};`
- Atribuição por índice:
 - `int matrizNome[2][2];`
 - `matrizNome[0][0] = 10;`

Alimentar toda matriz

- Repetição aninhada.

```
void main()
```

```
{
```

```
    int numeros[4][3]
```

```
    int i, j;
```

```
    for (i = 0; i < 4; i ++)
```

```
        for (j = 0; j < 3; j++)
```

```
            numeros[ i ][ j ] = i * j;
```

```
}
```

Listar toda matriz

- Repetição aninhada.

```
void main()
```

```
{
```

```
    int numeros[4][3]
```

```
    int i, j;
```

```
    .... // alimentou a matriz
```

```
    for (i = 0; i < 4; i ++)
```

```
        for (j = 0; j < 3; j ++)
```

```
            printf(" %d ",numeros[ i ][ j ]);
```

```
}
```

Exercício

- Faça este exemplo exibir uma matriz neste formato:

```
| 10  12 |
```

```
|  4   2 |
```



Exercício

- E se eu quiser trabalhar com um vetor de String?
- Faça um programa que leia uma lista de 10 nomes de professores.
- Imprima cada um deles e lembre-se de organizar seu código em funções e comentá-las.

Desafio

- Faça um jogo da velha e utilize matriz para resolver o problema.



Dúvidas?

alanamm.prof@gmail.com