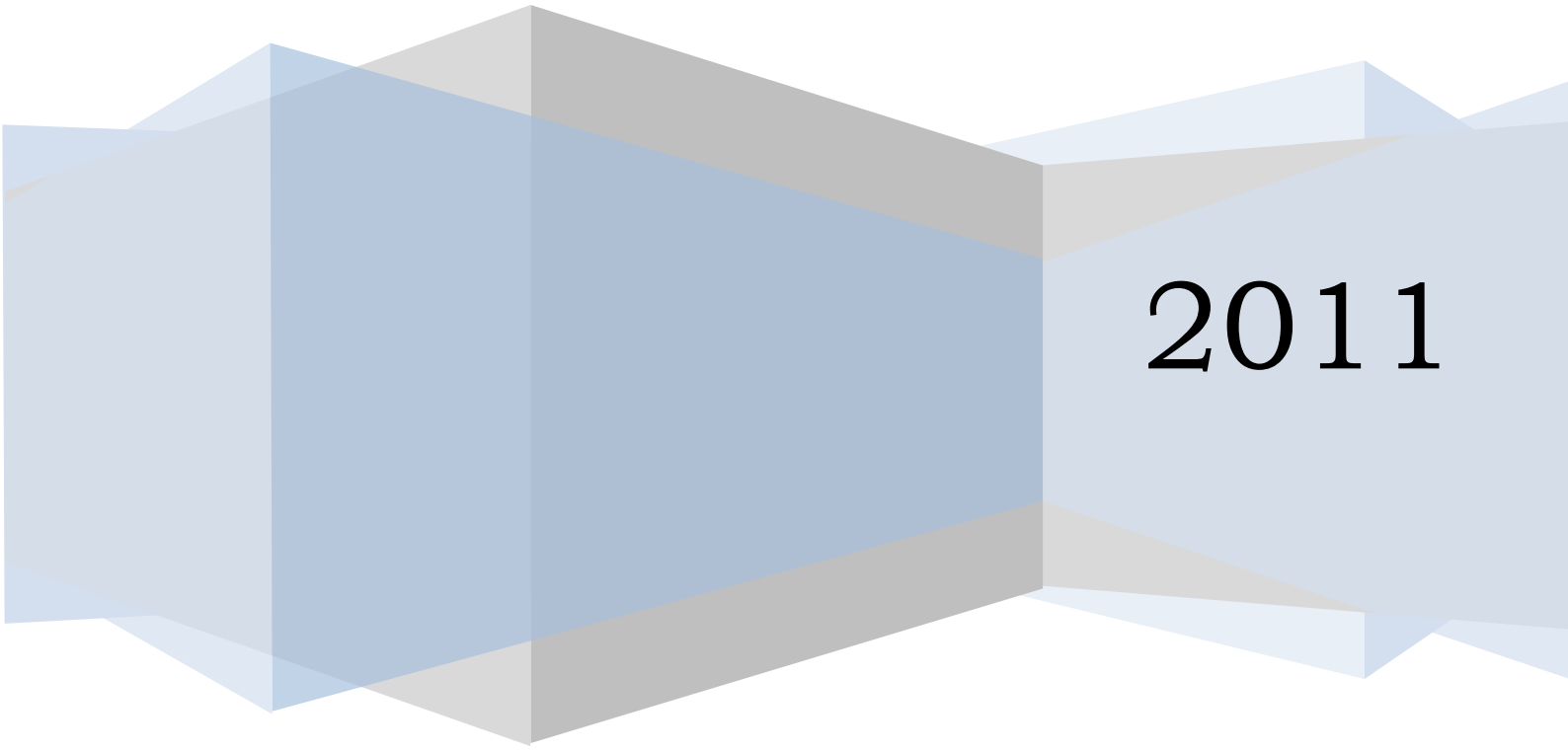
Flávio Augusto de Freitas
Introdução à Programação em Linguagem C/C++

http://flavioaf.blogspot.com

C/C++

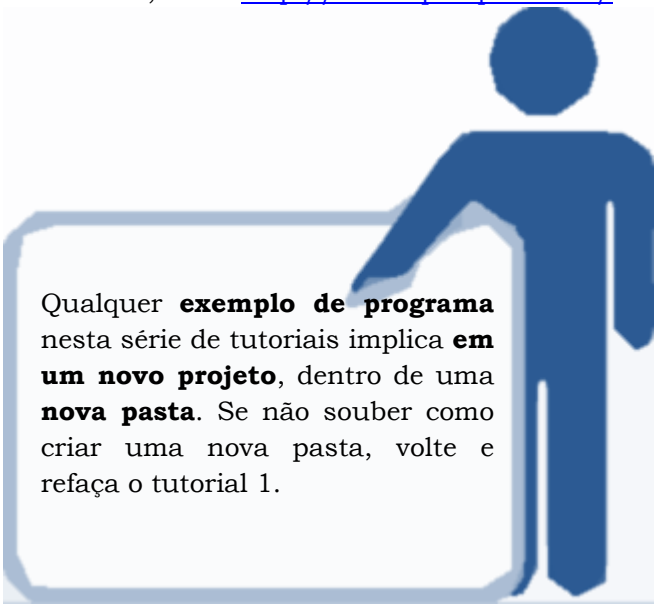
Tutorial 11 (usando Dev-C++ versão 4.9.9.2)



2011

1 INTRODUÇÃO

Esta série de tutoriais foi escrita usando o **Microsoft Windows 7 Ultimate** e o **Bloodshed Dev-C++** versão 4.9.9.2, que pode ser baixada em http://www.bloodshed.net. Se alguém quiser adquirir mais conhecimentos e quiser aprofundar no assunto, visite http://www.cplusplus.com/.



Qualquer **exemplo de programa** nesta série de tutoriais implica **em um novo projeto**, dentro de uma **nova pasta**. Se não souber como criar uma nova pasta, volte e refaça o tutorial 1.

2 O QUE SÃO ESTRUTURAS HOMOGÊNEAS?

Os **arrays** (vetores e matrizes) são estruturas de dados **homogêneas**, ou seja, são aglomerados de dados primitivos e todos de um **mesmo tipo**. Para se definir um aglomerado torna-se necessário especificar aquilo que é conhecido como sua **dimensão**, isto é, a quantidade de elementos que constituem o aglomerado e a **forma** como estarão dispostos. Aqui é feita apenas uma diferenciação didática entre arrays de uma, duas e três dimensões mas na verdade, conceitualmente, trata-se da mesma estrutura de dados.

Quando se fala em um array de uma dimensão (unidimensional) normalmente utiliza-se a palavra vetor. A expressão matriz geralmente é aplicada a arrays de mais de duas dimensões. A palavra matriz será usada de forma genérica para todos os tipos de arrays independentemente de sua dimensão. Esses nomes foram escolhidos dessa forma, pois já são de utilização consagrada no universo das linguagens de programação. A palavra vetor, em virtude de sua grande utilização na física/matemática, traz em si a idéia de uma reta ou uma linha, (algo unidimensional) e, da mesma forma, a palavra matriz, também em virtude de sua grande utilização na física/matemática, traz em si a ideia de uma tabela ou planilha (algo bidimensional), e é exatamente essa característica que nos permite realizar uma analogia com a forma como os dados são armazenados na memória do computador.

Cada um dos componentes de um aglomerado é dito um **elemento** desse aglomerado. Em um vetor ou matriz os elementos estão dispostos de forma **contígua** na memória e não há espaços vazios entre elementos adjacentes. Nos arrays, cada um dos elementos do aglomerado é referenciado por um mesmo identificador e individualizados por índices. Conforme a dimensão do array, são utilizados um, dois ou três índices para referenciar um dado elemento.

3 VETORES

Em linguagem C, declaramos vetores assim:

```
// vetor de inteiros com 10 posições
int A[10];

// vetor de ponto flutuante com 20 posições
float B[4, 5];

// string com 24 caracteres
char C[24];

// vetor de inteiros inicializado com valores
int vetor[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

// vetor de ponto flutuante inicializado com zeros
float S[3] = {0, 0, 0};

// vetor de inteiros longos não inicializado
long V[250];
```

Exemplo

Para uma turma de 5 alunos, com notas iguais a 8.5, 9.0, 7.0, 7.5 e 9.5, escreva um programa que calcule a média da turma. As notas devem estar alocadas em um vetor.

```
#include <cstdlib>
#include <iostream>

using namespace std;

const int ALUNOS = 5;

int main(int argc, char *argv[])
{
  double notas[ALUNOS] = {8.5, 9.0, 7.0, 7.5, 9.5};
  int pos = 0; // índice para elementos da matriz
  double media = 0; // guarda o valor da média
  double soma = 0; // acumula a soma das notas
```

```
  for (pos=0; pos <ALUNOS; pos++)
    // guarda a soma das notas
    soma = soma + notas[pos];

  media = soma / ALUNOS; // calcula e guarda a
média

  printf("Media = %f\n\n", media);

  system("PAUSE");
  return EXIT_SUCCESS;
}
```

4 EXERCÍCIOS RESOLVIDOS

1. Modifique o exemplo acima para que o usuário forneça o valor das notas;

```
#include <cstdlib>
#include <iostream>

using namespace std;

const int ALUNOS = 5;

int main(int argc, char *argv[])
{
  double notas[ALUNOS];
  int pos = 0; // índices para os elementos da
matriz
  double media = 0; // guarda o valor da
média
  double soma = 0; // acumula o somatório
das notas

  for (pos=0; pos <ALUNOS; pos++) {
    printf("Nota %d: ", pos + 1);
    scanf("%lf", &notas[pos]);
    printf("\n");
  }

  for (pos=0; pos <ALUNOS; pos++)
    soma = soma + notas[pos]; // guarda a
soma das notas

  media = soma / ALUNOS; // calcula e
guarda a média

  printf("Media = %f\n\n", media);

  system("PAUSE");
  return EXIT_SUCCESS;
}
```

2. Modifique o exemplo acima para que as notas sejam fornecidas aleatoriamente pelo programa;

```
#include <cstdlib>
#include <iostream>

using namespace std;

const int ALUNOS = 5;

int main(int argc, char *argv[])
{
  double notas[ALUNOS];
  int pos = 0; // índices para os elementos da
matriz
  double media = 0; // guarda o valor da
média
  double soma = 0; // acumula o somatório
das notas

  // inicializar o gerador de números
aleatórios
   srand(100);
  for (pos=0; pos <ALUNOS; pos++) {
    // para gerar números aleatórios de 0 a 10
    notas[pos] = rand() % 10;
    printf("Nota %d: %5.2f", pos + 1,
notas[pos]);
    printf("\n");
  }

  for (pos=0; pos <ALUNOS; pos++)
    soma = soma + notas[pos]; // guarda a
soma das notas

  media = soma / ALUNOS; // calcula e
guarda a média

  printf("Media = %5.2f\n\n", media);

  system("PAUSE");
  return EXIT_SUCCESS;
}
```

3. Altere o programa do item acima para classificar os elementos do vetor em ordem crescente e decrescente.

```
#include <cstdlib>
#include <iostream>

using namespace std;

const int ALUNOS = 5;

int main(int argc, char *argv[])
{
  double notas[ALUNOS];
  int pos = 0, i, j; // índices para os elementos
da matriz
```

```
  double aux;
  double media = 0; // guarda o valor da média
  double soma = 0; // acumula o somatório das notas

  // inicializar o gerador de números aleatórios
  // note que no programa anterior os números sempre
  // repetiam, agora o problema é evitado
  srand(time(NULL));

  printf("Notas nao ordenadas...\n");
  for (pos=0; pos <ALUNOS; pos++) {
    // para gerar números aleatórios de 0 a 10
    notas[pos] = rand() % 10;
    printf("Nota %d: %5.2f\n", pos + 1, notas[pos]);
  }

  printf("\nNotas em ordem crescente...\n");
  // método da bolha
  for (i=0; i <ALUNOS-1; i++)
    for (j=i+1; j <ALUNOS; j++) {
      if(notas[i] > notas[j]) {
        aux = notas[i];
        notas[i] = notas[j];
        notas[j] = aux;
      }
    }
  for (pos=0; pos <ALUNOS; pos++) {
    printf("Nota %d: %5.2f\n", pos + 1, notas[pos]);
  }

  printf("\nNotas em ordem decrescente...\n");
  // método da bolha
  for (i=0; i <ALUNOS-1; i++)
    for (j=i+1; j <ALUNOS; j++) {
      if(notas[i] < notas[j]) {
        aux = notas[i];
        notas[i] = notas[j];
        notas[j] = aux;
      }
    }
  for (pos=0; pos <ALUNOS; pos++) {
    printf("Nota %d: %5.2f\n", pos + 1, notas[pos]);
  }

  for (pos=0; pos <ALUNOS; pos++)
    soma = soma + notas[pos]; // guarda a soma das notas

  media = soma / ALUNOS; // calcula e guarda a média

  printf("\nMedia = %5.2f\n\n", media);

  system("PAUSE");
  return EXIT_SUCCESS;
}
```

5 Exercícios Propostos

a) Escreva um programa em C, para preencher um vetor V de 6 elementos com valores aleatórios entre 0 e 10, e que ordene este vetor da seguinte forma:
opção 1: ordena todos os elementos através do método bolha;
opção 2: ordena todos os elementos através do método bolha, mostrando passo a passo cada iteração (a cada clique do botão).
Observação: crie uma opção para gerar os números aleatórios para o vetor.

b) Escreva um programa em C que preencha um vetor V de 100 elementos com valores aleatórios entre 0 e 10 e localize neste vetor, um valor procurado pelo usuário. Se o elemento existir neste vetor, apresentar uma mensagem que o elemento foi encontrado, caso contrário apresentar uma mensagem que o elemento não foi encontrado.

c) Implemente modificações no método bolha para se diminuir a quantidade de comparações necessárias na ordenação, tornando-o mais eficiente.

d) Pesquisar sobre o tema "Pesquisa Binária" e desenvolver o programa;

6 Terminamos

Terminamos por aqui. O que está esperando, saia do Dev-C++ e corra para pegar o próximo tutorial em http://flavioaf.blogspot.com. Siga o blog, assim você fica sabendo das novidades no momento em que forem publicadas. Seguindo o blog você se mantém sempre atualizado de qualquer lançamento novo.