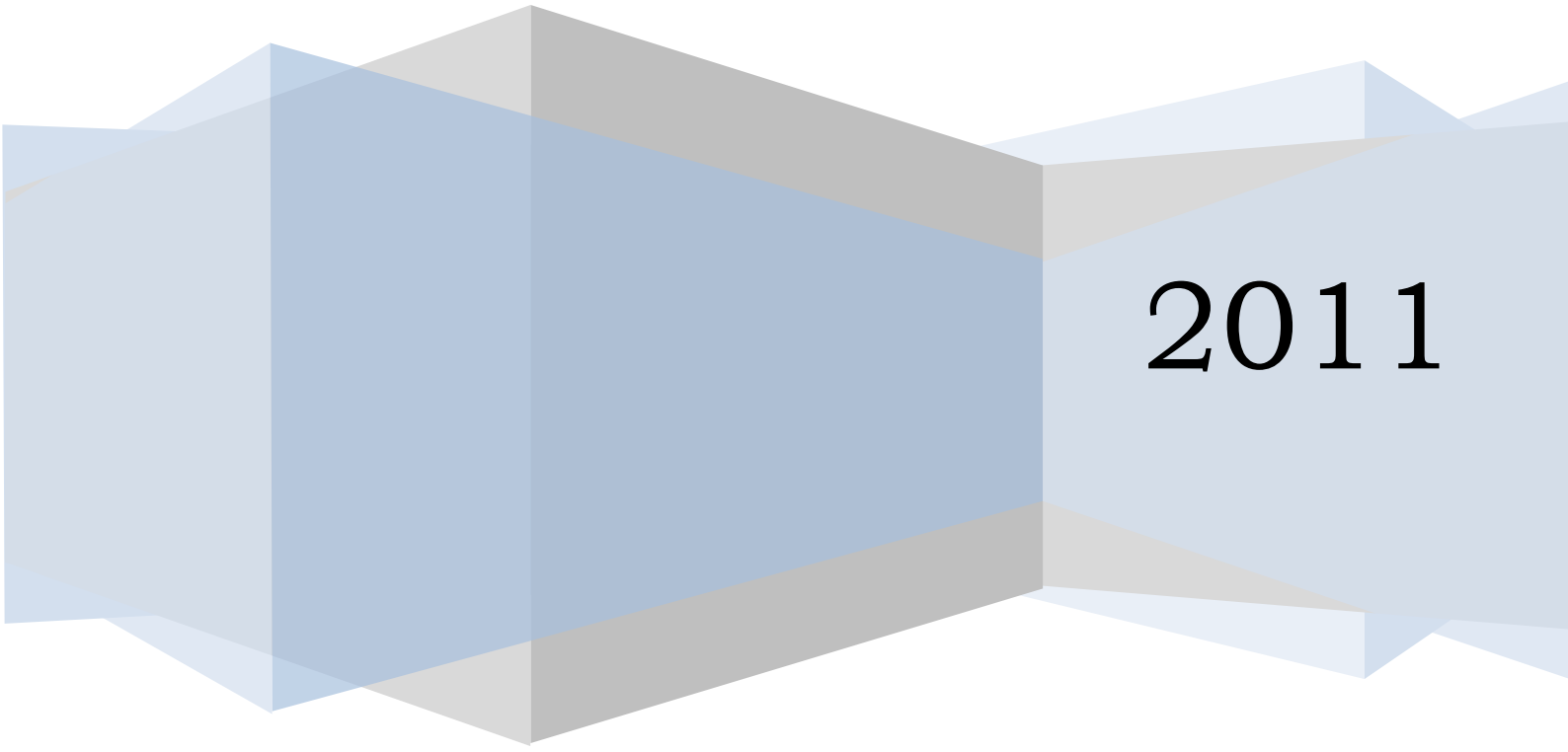
Flávio Augusto de Freitas
Introdução à Programação em Linguagem C/C++

http://flavioaf.blogspot.com

C/C++

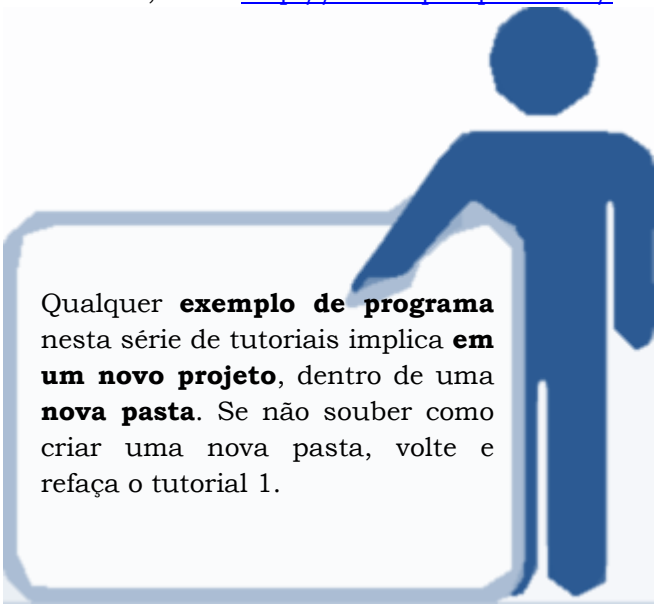
Tutorial 12 (usando Dev-C++ versão 4.9.9.2)



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?

Se você perdeu as definições do que são estruturas homogêneas, então pegue o tutorial 11 e faça-o antes de prosseguir.

3 EXERCÍCIOS RESOLVIDOS

1. Fazer um programa em C para ler uma quantidade N de alunos. Ler a nota de cada um dos N alunos e calcular a média aritmética das notas. Contar quantos alunos estão com a nota acima de 5.0.
   Observação: Se nenhum aluno tirou nota acima de 5.0, imprimir mensagem: Não há nenhum aluno com nota acima de 5.

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

using namespace std;

const int MAXALUNOS = 1000;

int main(int argc, char *argv[])
{
   float notas[MAXALUNOS];
   int i, lastNota, acima_da_media;
   float soma;
   printf("Forneca ate %d notas de alunos.\n", MAXALUNOS);
   printf("  Digite uma nota negativa para terminar.\n\n");

   for(i=0;i<MAXALUNOS;i++) {
      do {
         printf("Nota %d (entre 0 e 10): ", i + 1);
         scanf("%f", &notas[i]);
         if(notas[i] < 0.0) break; // sai do do-while
      } while (notas[i] > 10.0);
      printf("\n");

      if(notas[i] < 0.0) {
         lastNota = i; // guarda a última nota fornecida
         break; // sai do for
      }
   }

   printf("\n\nResultados:\n\n");
   soma = 0.0;
   acima_da_media = 0;
   for(i=0;i<lastNota;i++) {
      soma += notas[i];
      acima_da_media += notas[i] > 5 ? 1 : 0;
   }
   printf("Media das notas: %5.2f\n", soma / lastNota);
   if(acima_da_media > 0)
      printf("Alunos acima da media 5.0: %d\n\n", acima_da_media);
   else printf("Não ha nenhum aluno com nota acima de 5\n\n");

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

2. Elaborar um programa que gere um vetor com os primeiros 100 múltiplos de um número lido; depois, calcular a soma e a média do vetor.

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

using namespace std;

const int MAXMULTIPLOS = 100;

int main(int argc, char *argv[])
{
   int i, n, soma = 0;
   int multiplos[MAXMULTIPLOS];

   printf("Forneca um numero inteiro: ");
```

```
    scanf("%d", &n);
    printf("\n\n");

    for(i=0;i<MAXMULTIPLOS;i++)
      multiplos[i] = n * (i + 1);

    for(i=0;i<MAXMULTIPLOS;i++) {
      printf("%d ", multiplos[i]);
      soma += multiplos[i];
    }

    printf("\n\nSoma = %d\n", soma);
    // Na linha abaixo tivemos que usar casting,
    // pois as variáveis e constantes eram inteiras
    printf("Media = %f\n\n", (float) soma / (float) MAXMULTIPLOS);

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

3. Criar um programa que gere os primeiros 100 números primos.

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

using namespace std;

const int MAXPRIMES = 100;

int main(int argc, char *argv[])
{
    int primes[MAXPRIMES];
    int n, i, lastPrime = 0;
    int isPrime;

    printf("- 2 - ");
    primes[0] = 2;
    n = 3;
    while(lastPrime < MAXPRIMES) {
        isPrime = 1;
        for(i=0;i<=lastPrime;i++)
            if(n % primes[i] == 0) isPrime = 0;
        if(isPrime) {
            primes[++lastPrime] = n;
            printf("%d - ", n);
        }
        n += 2;
    }
    printf("\n\n");

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

4 Solução dos Exercícios Propostos do Tutorial 11

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 tecla pressionada).
Observação: crie uma opção para gerar os números aleatórios para o vetor.

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

using namespace std;

const int ALUNOS = 6;

int main(int argc, char *argv[])
{
  double notas[ALUNOS];
  int passo = 1, pos, i, j; // índices para os elementos da matriz
  double aux;
  bool passo_a_passo = false;
  int op;

  // 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("\nEscolha: \n");
  printf("1. Ordenar notas\n");
  printf("2. Ordenar notas passo a passo\n");
  printf("3. Sair\n\n");
  printf("Opcao: ");
  scanf("%d", &op);

  if (op == 3) return 0;
  if (op == 2) passo_a_passo = true;
```

```
  printf("\nNotas em ordem crescente...\n");
  // método da bolha
  for (i=0; i <ALUNOS-1; i++)
    for (j=i+1; j <ALUNOS; j++) {
      // mostra os passos da ordenação
      if(passo_a_passo) {
         printf("Passo %3d: ", passo++);
         for (pos=0; pos <ALUNOS; pos++)
            printf("%5.2f ", notas[pos]);
         printf("\n");
      }
      if(notas[i] > notas[j]) {
        aux = notas[i];
        notas[i] = notas[j];
        notas[j] = aux;
      }
    }

    // mostra o vetor finalmente ordenado
    if(passo_a_passo) {
       printf("Passo %3d: ", passo++);
       for (pos=0; pos <ALUNOS; pos++)
          printf("%5.2f ", notas[pos]);
       printf("\n");
    }

  printf("\nVetor ordenado...\n");
  for (pos=0; pos <ALUNOS; pos++) {
    printf("Nota %d: %5.2f\n", pos + 1,
notas[pos]);
  }

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

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.

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

using namespace std;

const int TAM = 100;

int main(int argc, char *argv[])
{
  double vetor[TAM];
  int pos; // índices para os elementos da
matriz
  double valor;

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

  for (pos=0; pos <TAM; pos++)
    // para gerar números aleatórios de 0 a 10
    vetor[pos] = rand() % 10;

  printf("Valor procurado: ");
  scanf("%lf", &valor);

  for (pos=0; pos <TAM; pos++) {
    if(vetor[pos]==valor)
      printf("Valor encontrado na posicao
%d\n", pos);
    else printf("Valor nao encontrado!!!\n");
  }

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

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

```
#include <cstdlib>
#include <iostream>
#include <math.h>

using namespace std;

const int TAM = 100;

int main(int argc, char *argv[])
{
  double vetor[TAM];
  int pos; // índices para os elementos da
matriz
  double aux;
  int i, j, troca;

  // inicializar o gerador de números
aleatórios
  srand(time(NULL));

  for (pos=0; pos <TAM; pos++)
    // para gerar números aleatórios de 0 a
100
    vetor[pos] = (fmod(rand(), 1000.0) + 1.0) /
100.0;
```

```
  troca = 1; // inicialmente, força indicação de troca
  // este for só executa se tiver havido troca no vetor
  for (i = TAM - 1; (i >= 1) && (troca == 1); i--) {
    troca = 0;
    for (j = 0; j < i; j++) {
      if (vetor[j] > vetor[j + 1]) {
        aux = vetor[j];
        vetor[j] = vetor[j + 1];
        vetor[j + 1] = aux;
        troca = 1; // informa que houve troca
      }
    }
  }

  for (pos=0; pos <TAM; pos++)
    printf("%5.3lf ", vetor[pos]);

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

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

5 Exercícios Propostos

a. Dados dois vetores, A (5 elementos) e B (8 elementos), faça um programa em C que imprima **todos** os **elementos comuns** aos dois vetores. Os vetores A e B deverão ter valores iniciais, por exemplo, A = {1, 2, 3, 4, 5, 6, 7} e B = {4, 5, 6, 7, 8, 9, 10}, assim, neste exemplo, o resultado do programa deve ser 4, 5, 6 e 7.
b. Faça um programa que determine o máximo e o mínimo de um conjunto de n números inteiros armazenados num vetor A de 10 elementos.
c. Durante uma corrida de automóveis com N voltas de duração foram anotados para um piloto, na ordem, os tempos registrados em cada volta. Fazer um programa em C para ler os tempos das N voltas, calcular e imprimir:
   • melhor tempo;
   • a volta em que o melhor tempo ocorreu;
   • tempo médio das N voltas;
d. Fazer um programa para ler um vetor de inteiros positivos e imprimir quantas vezes aparece o número 1, 3 e 4, nesta ordem. O vetor armazenará, no máximo, 100 valores. Sair do programa quando for digitado -1.

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.