```
//
// OBS.: RETIRAR OS COMENTARIOS /* */
//          PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
//

/*
// ------------------------------------ EXEMPLO101
                                // bibliotecas de funcoes auxiliares
#include <iostream>      // para cin e cout

using namespace std;

int main ( void )
{
 cout << "EXEMPLO101 - PRIMEIRO EXEMPLO EM C++";
 cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";

 getchar ( );                   // para esperar
 return    ( 0 );
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO102
                                // bibliotecas de funcoes auxiliares
#include <iostream>      // para cin e cout

using namespace std;

int main(void)
{
 system ( “cls” );              // ( ou “clear” ) para limpar a tela
                                // (dependente do sistema operacional)
 cout << "EXEMPLO102 - PRIMEIRO EXEMPLO EM C++";
 system ( “pause” );        // ( ou getch ( ) ) para esperar
                                // (dependente do sistema operacional)
 return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO103
                                // bibliotecas de funcoes auxiliares
#include <conio.h>       // para clrscr() e getch()
#include <iostream>      // para cin e cout

using namespace std;

int main ( void )
{
 clrscr ( );                    // funcao para limpar a tela
                                // (dependente do sistema operacional)
 cout << "EXEMPLO103 - PRIMEIRO EXEMPLO EM C++";
 cout << "\n";                  // para mudar de linha
 cout << "MATRICULA: ______ ALUNO : ____________________";
 cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
 getch ( );                     // para esperar
 return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO104
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
 cout << "EXEMPLO104 - PRIMEIRO EXEMPLO EM C++";
 cout <<  endl;               // para mudar de linha (="\n")
 cout << "MATRICULA: ______ ALUNO : ____________________";
 cout <<  endl;               // para mudar de linha
 cout << "PRESSIONAR <Enter> PARA TERMINAR.";
 getchar ( );                 // para esperar
 return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO105
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
 cout << "EXEMPLO105 - PRIMEIRO EXEMPLO EM C++";
 cout << "\nMATRICULA: ______ ALUNO : ____________________";
 cout << "\nEXEMPLOS DE VALORES : ";
 cout << "\nCARACTERE : "  << 'A';                  // letra ou simbolo
 cout << "\nINTEIRO       : "  << 10;                // valor sem parte fracionaria
 cout << "\nREAL          : "   << 3.1415;           // valor com parte fracionaria
 cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
 getchar ( );                 // para esperar
 return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO106
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout
#define   PI     3.1415     // definicao de macro (nome para substituir valor)

using namespace std;

int main ( void )
{
 cout << "EXEMPLO106 - PRIMEIRO EXEMPLO EM C++";
 cout << "\nMATRICULA: ______ ALUNO : ____________________";
 cout << "\nEXEMPLOS DE VALORES : ";
 cout << "\nCARACTERE : "  << 'A';       // letra ou simbolo
 cout << "\nINTEIRO       : "  << 10;       // valor sem parte fracionaria
 cout << "\nREAL          : "  << PI;       // emprego de macro
 cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
 getchar ( );                 // para esperar
 return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO107
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
                                // definicao de constante
 const float PI = 3.14;     // com nome e tipo (melhor)

 cout << "EXEMPLO107 - PRIMEIRO EXEMPLO EM C++";
 cout << "\nMATRICULA: ______ ALUNO : ___________________";
 cout << "\nEXEMPLOS DE VALORES : ";
 cout << "\nCARACTERE : "  << 'A';        // letra ou simbolo
 cout << "\nINTEIRO       : "  << 10;       // valor sem parte fracionaria
 cout << "\nREAL          : "  << PI;       // constante real
 cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
 getchar ( );                   // para esperar
 return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO108
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
                                // definicao de constante
 const float PI = 3.14;     // com nome e tipo (melhor)
                                // definicao de variavel real
 float   X = 10.01;             // com atribuicao de valor inicial

 cout << "EXEMPLO108 - PRIMEIRO EXEMPLO EM C++";
 cout << "\nMATRICULA: ______ ALUNO : ___________________";
 cout << "\nEXEMPLOS DE VALORES : ";
 cout << "\nCARACTERE : "  << 'A';        // letra ou simbolo
 cout << "\nINTEIRO       : "  << 10;       // valor sem parte fracionaria
 cout << "\nREAL          : "  << PI;       // constante real
 cout << "\nREAL          : "  << X;        // variavel real

 cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
 getchar ( );                   // para esperar
 return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO109
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
                                // definicao de constante
 const float PI = 3.14;     // com nome e tipo (melhor)
                                // definicao de variavel real
 float X = 10.01;
                                // definicao de variavel inteira
 int   I = 10;

 cout << "EXEMPLO109 - PRIMEIRO EXEMPLO EM C++";
 cout << "\nMATRICULA: ______ ALUNO : ____________________";
 cout << "\nEXEMPLOS DE VALORES : ";
 cout << "\nINTEIRO : "  <<  I ;
 cout << "\nREAL      : "  << X ;
 cout << "\nREAL      : "  << PI;
 cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
 getchar ( );                   // para esperar
 return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO110
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
                                // definicao de constante
 const float PI = 3.14;     // com nome e tipo (melhor)
                                // definicao de variavel real
 float X = 10.01;
                                // definicao de variavel inteira
 int   I = 10;
                                // definicao de variavel caractere
 char  N = '\n';                // mudar de linha

 cout << "EXEMPLO110 - PRIMEIRO EXEMPLO EM C++";
 cout << "\nMATRICULA: ______ ALUNO : ____________________";
 cout << N << "EXEMPLOS DE VALORES : ";
 cout << N << "INTEIRO : "  <<  I ;
 cout << N << "REAL      : "  << X ;
 cout << N << "REAL      : "  << PI;
 cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
 getchar ( );                   // para esperar
 return EXIT_SUCCESS;
} // fim do programa
*/
```

```
//
// OBS.: RETIRAR OS COMENTARIOS /* */
//          PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
//

/*
// ------------------------------------ EXEMPLO201
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR UM VALOR INTEIRO
// VARIAVEL:
   int X;

   cout << "EXEMPLO201 - LER E IMPRIMIR UM VALOR INTEIRO";
   cout << "\nFORNECER UM VALOR INTEIRO QUALQUER: ";
   cin  >>  X;
   cout << "\nO VALOR DIGITADO FOI: " << X;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO202
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR UM VALOR REAL
// VARIAVEL:
   float X;

   cout << "EXEMPLO202 - LER E IMPRIMIR UM VALOR REAL";
   cout << "\nFORNECER UM VALOR REAL QUALQUER: ";
   cin   >>  X;
   cout << "\nO VALOR DIGITADO FOI: " << X;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO203
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR UM CARACTERE
// VARIAVEL:
   char X;

   cout << "EXEMPLO203 - LER E IMPRIMIR UM CARACTERE";
   cout << "\nFORNECER UM CARACTERE QUALQUER: ";
   cin   >>  X;
   cout << "\nO VALOR DIGITADO FOI: " << X;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO204
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR, NO MAXIMO, 10 CARACTERES
// VARIAVEL:
   char X [10];

   cout << "EXEMPLO204 - LER E IMPRIMIR, NO MAXIMO, 09 CARACTERES";
   cout << "\nDIGITE, NO MAXIMO, 09 CARACTERES QUAISQUER: ";
   cin   >>  X;
   cout << "\nFOI DIGITADO: " << X;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO205
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E SOMAR DOIS VALORES INTEIROS
// VARIAVEIS:
   int  X,Y,Z;

   cout << "EXEMPLO205 - LER E SOMAR DOIS VALORES INTEIROS";
   cout << "\nFORNECER UM VALOR INTEIRO QUALQUER: ";
   cin   >>  X;
   cout << "\nFORNECER OUTRO VALOR INTEIRO QUALQUER: ";
   cin   >>  Y;
   Z = X + Y;
   cout << "\nA SOMA DOS DOIS = " << Z;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO206
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E SUBTRAIR DOIS VALORES REAIS
// VARIAVEIS:
   float X,Y,Z;

   cout << "EXEMPLO206 - LER E SUBTRAIR DOIS VALORES REAIS";
   cout << "\nFORNECER UM VALOR REAL QUALQUER: ";
   cin   >>  X;
   cout << "\nFORNECER OUTRO VALOR REAL QUALQUER: ";
   cin   >>  Y;
   Z = X - Y;
   cout << "\nA DIFERENCA ENTRE OS DOIS = " << Z;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO207
                                // bibliotecas de funcoes auxiliares
#include <bool.h>          // para valores logicos
#include <iostream>        // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA OPERAR VALORES LOGICOS
// VARIAVEIS:
   bool X,Y,Z;

   cout << "EXEMPLO207 - OPERAR VALORES LOGICOS";
   X = TRUE;
   Y = FALSE;
   Z = X || Y;                 // X ou Y
   cout << "\nA DISJUNCAO ENTRE VERDADEIRO E FALSO = " << Z;
   Z = X && Y;                 // X e  Y
   cout << "\nA CONJUNCAO ENTRE VERDADEIRO E FALSO = " << Z;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO208
                                // bibliotecas de funcoes auxiliares
#include <iostream>        // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA CALCULAR A VELOCIDADE DE UM VEICULO
// VARIAVEIS:
   float    D,                 // Distancia
            T,                 // Tempo
            V;                 // Velocidade

   cout << "EXEMPLO208 - CALCULAR A VELOCIDADE DE UM VEICULO";
   cout << "\nFORNECER UMA DISTANCIA QUALQUER EM METROS: ";
   cin   >>  D;
   cout << "\nFORNECER O TEMPO PARA PERCORRE-LA EM SEGUNDOS: ";
   cin   >>  T;
   V = D / T;
   cout << "\nV = D / T = " << V << " m/s ";
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO209
                                // bibliotecas de funcoes auxiliares
#include <string.h>        // para lidar com caracteres
#include <iostream>        // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA COMPARAR CARACTERES COM UMA SENHA
// CONSTANTE:
   const char SENHA[5] = "XXXX";
// VARIAVEL:
   char S [10];

   cout << "EXEMPLO209 - COMPARAR CARACTERES COM UMA SENHA";
   cout << "\nFORNECER UMA CADEIA DE CARACTERES QUALQUER: ";
   cin   >>  S;
   cout << "\nA COMPARACAO COM A SENHA = " << (strcmp(S,SENHA)==0)?1:0;
//               strcmp(S1,S2) compara S1 com S2
//               igual a 0: S1 =  S2 => 1 ( verdadeiro )
//               diferente: S1 <> S2 => 0 (   falso    )
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO210
                                // bibliotecas de funcoes auxiliares
#include <math.h>          // para operacoes matematicas
#include <iostream>        // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA CALCULAR O ARCO TRIGONOMETRICO DE UM SENO
// CONSTANTE:
 const float PI = 3.14;
// VARIAVEIS:
   float ARCO,
         COSSENO,
         SENO,
         TANGENTE;

   cout << "EXEMPLO210 - CALCULAR O ARCO TRIGONOMETRICO DE UM SENO";
   cout << "\nFORNECER O VALOR DO SENO: ";
   cin   >>  SENO;
   COSSENO  = sqrt( 1.0 - pow(SENO,2));
   TANGENTE = SENO / COSSENO;
   ARCO         = atan(TANGENTE);
   cout << "\nO ARCO TRIGONOMETRICO EM GRAUS = " << (ARCO*180.0/PI);
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
//
// OBS.: RETIRAR OS COMENTARIOS /* */
//          PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
//

/*
// ------------------------------------ EXEMPLO301
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER UM VALOR INTEIRO E VERIFICAR SE E' ZERO
// VARIAVEL:
   int X;

   cout << "EXEMPLO301 - LER E TESTAR UM VALOR INTEIRO";
   cout << "\nFORNECER UM VALOR INTEIRO QUALQUER: ";
   cin   >>  X;
   if( X == 0 )
     cout << "\nO VALOR DIGITADO FOI ZERO";
   else
     cout << "\nO VALOR DIGITADO NAO FOI ZERO";
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO302
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER UM REAL E TESTAR SE DIFERENTE DE ZERO
// VARIAVEL:
   float X;

   cout << "EXEMPLO302 - LER E TESTAR UM VALOR REAL";
   cout << "\nFORNECER UM VALOR REAL DIFERENTE DE ZERO: ";
   cin   >>  X;
   if( X != 0.0 )
     cout << "\nO VALOR DIGITADO FOI DIFERENTE DE ZERO";
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO303
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER CARACTERE E VERIFICAR SE E' UM ALGARISMO
// VARIAVEL:
   char X;

   cout << "EXEMPLO303 - LER E TESTAR UM CARACTERE";
   cout << "\nFORNECER UM ALGARISMO QUALQUER: ";
   cin   >>  X;
   if( X >= '0' && X <= '9' )
   {
     cout << "\nO VALOR DIGITADO FOI UM ALGARISMO";
     cout << "\nO ALGARISMO DIGITADO FOI: " << X ;
   } // if ALGARISMO
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO304
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER CARACTERE E TESTAR SE NAO E' ALGARISMO
// VARIAVEL:
   char X;

   cout << "EXEMPLO304 - LER E TESTAR CARACTERE";
   cout << "\nFORNECER UM CARACTERE QUALQUER: ";
   cin   >>  X;
   if( !( X >= '0' && X <= '9') )
   {
     cout << "\nNAO FOI DIGITADO UM ALGARISMO";
     cout << "\nFOI DIGITADO O CARACTERE: " << X;
   } // if NAO ALGARISMO
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO305
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E TESTAR A IGUALDADE DE DOIS INTEIROS
// VARIAVEIS:
   int X,Y;

   cout << "EXEMPLO305 - LER E TESTAR DOIS VALORES INTEIROS";
   cout << "\nFORNECER UM VALOR INTEIRO QUALQUER: ";
   cin   >>  X;
   cout << "\nFORNECER OUTRO VALOR INTEIRO QUALQUER: ";
   cin   >>  Y;
   if( X == Y )
     cout << "\nDOIS VALORES IGUAIS";
   else
   {
     cout << X;
     cout << " DIFERENTE DE ";
     cout << Y;
   }
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO306
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E TESTAR DOIS VALORES REAIS
// VARIAVEIS:
   float X,Y;

   cout << "EXEMPLO306 - LER E TESTAR DOIS VALORES REAIS";

   cout << "\nFORNECER UM VALOR REAL QUALQUER: ";
   cin   >>  X;
   cout << "\nFORNECER OUTRO VALOR REAL QUALQUER: ";
   cin  >>   Y;
   if( !(X == Y) )
   {
     cout << X;
     cout << " DIFERENTE DE ";
     cout << Y;
   }
   else
   {
     cout << "VALORES IGUAIS";
   } // if VALORES DIFERENTES
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                  // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO307
                                // bibliotecas de funcoes auxiliares
#include <bool.h>          // para variaveis logicas
#include <iostream>        // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA TRATAR ALTERNATIVAS COM VALORES LOGICOS
// VARIAVEIS:
   int  X,Y;
   bool Z;

   cout << "EXEMPLO307 - TRATAR VALORES LOGICOS";
   cout << "\nFORNECER UM VALOR INTEIRO QUALQUER: ";
   cin   >>  X;
   cout << "\nFORNECER OUTRO VALOR INTEIRO QUALQUER: ";
   cin   >>  Y;
   Z = (X == Y);
   if( Z )
     cout << "VALORES IGUAIS";
   else
     cout << "VALORES DIFERENTES";
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO308
                                // bibliotecas de funcoes auxiliares
#include <iostream>        // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E TESTAR UMA LETRA
// VARIAVEL:
   char X;

   cout << "EXEMPLO308 - LER E TESTAR UMA LETRA";
   cout << "\nFORNECER UMA LETRA QUALQUER: ";
   cin   >>  X;
   If( X >= 'A' && X <= 'Z' )
     cout << "FOI DIGITADA UMA LETRA MAIUSCULA";
   else
     if( X >= 'a' && X <= 'z' )
       cout << "FOI DIGITADA UMA LETRA MINUSCULA";
     else
       cout << "NAO FOI DIGITADA UMA LETRA";
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO309
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA COMPARAR CARACTERES < , = , >
// CONSTANTES:
   const char MAIOR = '>',
              IGUAL = '=',
              MENOR = '<';
// VARIAVEL:
   char X;

   cout << "EXEMPLO309 - COMPARAR CARACTERES < , = , >";
   cout << "\nFORNECER UM DOS CARACTERES CITADOS: ";
   cin   >>  X;
   switch( X )
   {
     case MAIOR:   cout << "FOI DIGITADO O SINAL DE MAIOR";
      break;
     case IGUAL:    cout << "FOI DIGITADO O SINAL DE IGUAL";
      break;
     case MENOR:  cout << "FOI DIGITADO O SINAL DE MENOR";
      break;
     default  :  cout << "FOI DIGITADO UM OUTRO CARACTERE QUALQUER";
   } // COMPARACAO DE X COM < , = , >
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO310
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA IDENTIFICAR CARACTERES
// VARIAVEL
   char X;

   cout << "EXEMPLO310 - IDENTIFICAR CARACTERES";
   cout << "\nFORNECER UM CARACTERE QUALQUER: ";
   cin   >>  X;
   switch ( X )
   {
     case 'A':
     case 'E':
     case 'I':
     case 'O':
     case 'U': cout << "FOI DIGITADO UMA VOGAL";
       break;

    case '0':
    case '1':
    case '2':
    case '3':
    case '4':
    case '5':
    case '6':
    case '7':
    case '8':
    case '9': cout << "FOI DIGITADO UM ALGARISMO";
              cout << "\nO NUMERO CORRESPONDENTE = " << (X-48);
     break;

    default: cout << "FOI DIGITADO UM OUTRO CARACTERE QUALQUER";
   } // IDENTIFICACAO DE UM CARACTERE
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
//
// OBS.: RETIRAR OS COMENTARIOS /* */
//           PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
//

/*
// ------------------------------------ EXEMPLO401
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR 03 VALORES INTEIROS
// VARIAVEIS :
   int X,
      CONTADOR;

   cout << "EXEMPLO401 - LER E IMPRIMIR 03 VALORES INTEIROS";
   cout <<  endl;              // mudar de linha
   CONTADOR = 1;
   while( CONTADOR <= 3 ) // REPETIR
   {
     cout <<  endl;                      // mudar de linha
     cout <<  CONTADOR << " FORNECER UM VALOR INTEIRO : ";
     cin  >>  X;
     cout << "\nO VALOR DIGITADO FOI : " << X << endl;
     CONTADOR = CONTADOR + 1;
   }                              // ENQUANTO CONTADOR <= 3

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                   // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO402
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout
using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
// VARIAVEIS :
 int X, N, CONTADOR;

   cout << "EXEMPLO402 - LER E IMPRIMIR (N) VALORES INTEIROS\n";
   cout << "\nFORNECER O NUMERO DE VEZES (N) : ";
   cin   >>  N;
   CONTADOR = 1;
   while( CONTADOR <= N )
   {
     cout << "\n" << CONTADOR;
     cout << " FORNECER UM VALOR INTEIRO QUALQUER : ";
     cin  >>  X;
     cout << "\nO VALOR DIGITADO FOI : " << X << endl;
     CONTADOR = CONTADOR + 1;
   } // ENQUANTO CONTADOR <= N

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO403
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout
using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
// VARIAVEIS :
   int X, N;

   cout << "EXEMPLO403 - LER E IMPRIMIR (N) VALORES INTEIROS\n";
   cout << "\nFORNECER O NUMERO DE VEZES (N) : ";
   cin   >>  N;
   while( N > 0 )               // REPETIR
   {
     cout << "\n" << N;
     cout << " FORNECER UM VALOR INTEIRO QUALQUER : ";
     cin   >>  X;
     cout << "\nO VALOR DIGITADO FOI : " << X << endl;
     N = N - 1;
   }                            // ENQUANTO N > 0
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO404
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR 03 VALORES INTEIROS
// VARIAVEIS :
   int X,
       CONTADOR;

   cout << "EXEMPLO404 - LER E IMPRIMIR 03 VALORES INTEIROS\n";
   for( CONTADOR = 1; CONTADOR <= 3; CONTADOR = CONTADOR+1)
   {
     cout << "\n" << CONTADOR << " FORNECER UM VALOR INTEIRO : ";
     cin   >>  X;
     cout << "\nO VALOR DIGITADO FOI : " << X << endl;
   } // PARA CONTADOR EM [1:3]

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO405
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
// VARIAVEIS :
   int X,
       N,
       CONTADOR;

   cout << "EXEMPLO405 - LER E IMPRIMIR (N) VALORES INTEIROS\n";
   cout << "\nFORNECER O NUMERO DE VEZES (N) : ";
   cin   >>  N;
   for( CONTADOR = 1; CONTADOR <= N; CONTADOR++)
   {
     cout << "\n" << CONTADOR << " FORNECER UM VALOR INTEIRO : ";
     cin   >>  X;
     cout << "\nO VALOR DIGITADO FOI : " << X << endl;
   } // PARA CONTADOR EM [1:N]

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO406
                                // bibliotecas de funcoes auxiliares
#include <iostream>      // para cin e cout
using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR 03 VALORES INTEIROS
// VARIAVEIS :
   int X,
      CONTADOR;

   cout << "EXEMPLO406 - LER E IMPRIMIR 03 VALORES INTEIROS\n";
   CONTADOR = 1;
   do                           // REPETIR
   {
     cout << "\n" << CONTADOR << " DIGITE UM VALOR INTEIRO : ";
     cin   >>  X;
     cout << "\nO VALOR DIGITADO FOI : " << X << endl;
     CONTADOR = CONTADOR + 1;
   }
   while( CONTADOR <= 3 );// ATE' ( CONTADOR > 3 )
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO407
                                // bibliotecas de funcoes auxiliares
#include <iostream>      // para cin e cout
using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
// VARIAVEIS :
   int X,
      CONTADOR;

   cout << "EXEMPLO407 - LER E IMPRIMIR (N) VALORES INTEIROS\n";
   cout << "\nFORNECER O NUMERO DE VEZES (N) : ";
   cin   >>  CONTADOR;
   do                           // REPETIR
   {
     cout << "\n" << CONTADOR << " FORNECER UM VALOR INTEIRO : ";
     cin   >>  X;
     cout << "\nO VALOR DIGITADO FOI : " << X << endl;
     CONTADOR = CONTADOR - 1;
   }
   while( CONTADOR > 0 ); // ATE' ( CONTADOR <= 3 )
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO408
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER E IMPRIMIR INTEIROS DIFERENTES DE ZERO
// VARIAVEL :
   int X;

   cout << "EXEMPLO408 - LER E IMPRIMIR INTEIROS NAO NULOS\n";
   cout << "\nFORNECER UM VALOR INTEIRO (0 = PARAR) : ";
   cin   >>  X;
   while( X != 0 )             // REPETIR
   {
     cout << "\nO VALOR DIGITADO FOI : " << X << endl;
     cout << "\nDIGITE UM VALOR INTEIRO QUALQUER : ";
     cin   >>  X;
   }                           // ENQUANTO X DIFERENTE DE ZERO

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO409
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER UM INTEIRO DIFERENTE DE ZERO
// VARIAVEL :
   int X;

   cout << "EXEMPLO409 - PARA LER UM INTEIRO NAO NULO\n";
   cout << "\nFORNECER UM VALOR DIFERENTE DE ZERO : ";
   cin   >>  X;
   while( X == 0 )                     // REPETIR
   {
      cout << "\nFORNECER UM VALOR DIFERENTE DE ZERO : ";
      cin   >>  X;
   }                           // ENQUANTO X IGUAL A ZERO
   cout << "\nDIGITADO UM NUMERO DIFERENTE DE ZERO\n";

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO410
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int main ( void )
{
// PROGRAMA PARA LER UM INTEIRO DIFERENTE DE ZERO
// VARIAVEL :
   int X;

   cout << "EXEMPLO410 - LER UM INTEIRO NAO NULO\n";
   do                           // REPETIR
   {
     cout << "\nFORNECER UM VALOR DIFERENTE DE ZERO : ";
     cin   >>  X;
   }
   while( X == 0 );                      // ATE' X DIFERENTE DE ZERO
   cout << "\nDIGITADO UM NUMERO DIFERENTE DE ZERO\n";

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
//
// OBS.: RETIRAR OS COMENTARIOS /* */
//       PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
//

/*
// ------------------------------------ EXEMPLO501
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

void P1( void )
{
   cout <<   endl;
   cout << "\nCHAMADO O PROCEDIMENTO P1 SEM PARAMETROS";
   cout <<   endl;
} // fim procedimento P1( )

int main ( void )
{
// PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

   cout <<  "EXEMPLO0501 - CHAMADA A UM PROCEDIMENTO";
   P1( );                       // chamada ao procedimento
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO502
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int X; // VARIAVEL GLOBAL

void P1 ( void )
{
 cout <<  endl;
 cout << "CHAMADO O PROCEDIMENTO P1 " << X << " VEZES";
 cout <<  endl;
} // fim procedimento P1( )

int main ( void )
{
// PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

  cout <<  "EXEMPLO0502 - CHAMADA COM VARIAVEL GLOBAL\n";
   for ( X = 1; X <= 5; X = X + 1 )
     P1( );                             // chamar 5 vezes
   cout <<  endl;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
```

```
// ------------------------------------ EXEMPLO503
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int X; // VARIAVEL GLOBAL

void P1 ( void )
{
   X = X + 1;
   cout << "\nCHAMADO O PROCEDIMENTO P1 " << X << " VEZES";
   if ( X < 5 )   P1( );                        // chamar recursivamente
    cout << "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " << X;
   X = X - 1;
   getch( );                    // para esperar
} // fim procedimento P1( )

int main ( void )
{
// PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

   cout <<  "EXEMPLO0503 - CHAMADA/RETORNO COM VARIAVEL GLOBAL\n";
   X = 0;
   P1 ( ); // OBSERVAR A RECURSIVIDADE !
   cout <<  endl;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO504
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

void P1 (int X)
{
   cout << "\nCHAMADO O PROCEDIMENTO P1 " << X << " VEZES";
   if ( X < 5 )
     P1( X + 1 );               // chamar recursivamente com parametro
   cout << "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " << X;
   getch( );                    // para esperar
} // fim procedimento P1( )

int main ( void )
{
// PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

   cout <<  "EXEMPLO0904 - CHAMADA/RETORNO COM PARAMETRO\n";
   P1 ( 1 ); // OBSERVAR REPETICAO FINITA, SEM VARIAVEL GLOBAL !
   cout <<  endl;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO505
                                // bibliotecas de funcoes auxiliares
#include <iostream>      // para cin e cout

using namespace std;

void P1 (int X)
{
   cout << "\nCHAMADO O PROCEDIMENTO P1 " << X << " VEZES";
   if ( X > 1 )
     P1 ( X – 1 );
   cout << "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " << X;
   getchar ( );                 // para esperar
} // fim procedimento P1( )

int main ( void )
{
// PROGRAMA PARA MOSTRAR PASSAGEM DE PARAMETRO POR VALOR

   cout <<  "EXEMPLO0505 - CHAMADA/RETORNO COM PARAMETRO\n";
   P1 ( 5 ); // OBSERVAR REPETICAO FINITA, SEM VARIAVEL GLOBAL !
   cout <<  endl;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO506
                                // bibliotecas de funcoes auxiliares
#include <iostream>      // para cin e cout

using namespace std;

void P2 (int X); // PROTOTIPO DE PROCEDIMENTO

void P1 (int X)
{
   cout << "\nCHAMADO O PROCEDIMENTO P1 COM X = " << X ;
   if ( X < 5 )
     P2 ( X );
   cout << "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " << X;
   getchar ( );                 // para esperar
} // fim do procedimento P1( )

void P2 (int X)
{
   cout << "\nCHAMADO O PROCEDIMENTO P2 COM X = " << X ;
   X = X+1;
   cout << "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " << X;
   getchar ( );                 // para esperar
   P1( X );
} // fim do procedimento P2( )
```

```
int main ( void )
{
// PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

   cout <<  "EXEMPLO0506 - CHAMADA/RETORNO COM PARAMETRO\n";
   P1 ( 1 ); // OBSERVAR RECURSIVIDADE INDIRETA !
   cout <<  endl;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO507
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

void P1 (int &I)
{
   X = X + 1;
   cout << "\nCHAMADO O PROCEDIMENTO P1 " << X << " VEZES";
   if ( X < 5 )
     P1 ( X );
   cout << "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " << X;
   X = X - 1;
   getchar ( );                 // para esperar
} // fim procedimento P1( )

int main ( void )
{
// PROGRAMA PARA MOSTRAR PASSAGEM DE PARAMETRO POR REFERENCIA
// VARIAVEL LOCAL
   int X;

   cout <<  "EXEMPLO0507 - CHAMADA/RETORNO COM REFERENCIA\n";
   X = 0;
   P1 ( X );         // OBSERVAR REPETICAO FINITA !
   cout <<  endl;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO508
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

void P2 (int X); // PROTOTIPO DE PROCEDIMENTO

void P1 (int X)
{
   X = X + 1;
   cout << "\nCHAMADO O PROCEDIMENTO P1 " << X << " VEZES";
   if ( X < 4 )
   {
     P1 ( X );
     P2 ( X );
   }
   cout <<   endl;
   cout << "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " << X;
   getchar ( ) ;                // para esperar
} // fim do procedimento P1( )

void P2 (int X)
{
   cout << "\nCHAMADO O PROCEDIMENTO P2 " << X << " VEZES";
   if ( X > 1 )
     P2 ( X – 1 );
   cout << "\nRETORNANDO AO PROCEDIMENTO P2 PARA A CHAMADA " << X;
   getchar ( );                 // para esperar
} // fim procedimento P2( )

int main ( void )
{
// PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

   cout <<  "EXEMPLO0508 - MULTIPLAS CHAMADAS/RETORNOS\n";
   P1 ( 1 );
   cout <<  endl;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
- Digitar o conteudo abaixo em um arquivo com o nome LIB01.HPP :

// ------------------------------------ EXEMPLO509a
                                // bibliotecas de funcoes auxiliares
#include <conio.h>          // para clrscr ( ) e getchar ( )

 void LIMPAR_TELA ( void )
 { clrscr ( ); }

 void ESPERAR ( void )
 { getchar ( ); }

- Digitar o conteudo abaixo em outro arquivo com o nome LIB02.HPP :

// ------------------------------------ EXEMPLO509b
                                // bibliotecas de funcoes auxiliares
#include <iostream>         // para cin e cout

using namespace std;

 void ESCREVER_CARACTERES ( char S[ ] )
 { cout << S; }

- Digitar o conteudo abaixo em um terceiro arquivo :

/*
// ------------------------------------ EXEMPLO509
                                // bibliotecas de funcoes auxiliares
#include "lib01.hpp"        // para limpar a tela e esperar
#include "lib02.hpp"        // para escrever caracteres

int main ( void )
{
// PROGRAMA PARA MOSTRAR O USO DE MODULOS

   LIMPAR_TELA ( );
   ESCREVER_CARACTERES("EXEMPLO0509 - USO DE MODULOS\n");

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR.";
   ESPERAR ( );
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
- Digitar o conteudo abaixo em um arquivo com o nome LIB03.HPP :

// CONSTANTES GLOBAIS :

   const int MINX  =   1;  // minimo valor para coluna
   const int MAXX = 80;  // maximo valor para coluna
   const int MINY  =   1;  // minimo valor para linha
   const int MAXY = 24;  // maximo valor para linha

// VARIAVEIS GLOBAIS :

   char  C  = ' ';             // caractere lido
   int     X  = 40,            // coluna
           Y = 12;             // linha
```

```
// PROCEDIMENTOS E FUNCOES :

int PARAR ( )                // condicao de parada igual a '.'
{ return (C == '.'); }

void MOVER_CURSOR( void )
{
   do                 // repetir
   {
    gotoxy (X,Y);             //  posicionar em uma coluna e linha
    C = getchar ( );          //  ler caractere
    switch(C)                 //  escolher o movimento
    {
      case '8' : Y = Y  - 1; break;    // mover para baixo
      case '2' : Y = Y + 1; break;    // mover para cima
      case '4' : X = X  - 1; break;    // mover para a esquerda
      case '6' : X = X + 1; break;    // mover para a direita
    }                         //  fim escolher
   }                   // ate' parar
   while( ! PARAR( ) );
} // fim procedimento MOVER_CURSOR( )
```

- Digitar o conteudo abaixo em outro arquivo :

```
/*
// ------------------------------------ EXEMPLO510
                              // bibliotecas de funcoes auxiliares
#include "lib01.hpp"       // para limpar a tela e esperar
#include "lib02.hpp"       // para escrever caracteres
#include "lib03.hpp"       // para constantes, variaveis e outros

int main ( void )
{
// PROGRAMA PARA MOSTRAR O USO DE MODULOS

   LIMPAR_TELA ( );
   ESCREVER_CARACTERES ( "EXEMPLO0510 - USO DE MODULOS\n" );
   ESCREVER_CARACTERES ("PARA PARAR, APERTAR PONTO ('.')\n");
   ESCREVER_CARACTERES
                          ("USAR O TECLADO NUMERICO PARA MOVER O CURSOR\n");

   while( ! PARAR( ) )
     MOVER_CURSOR ( );

   return EXIT_SUCCESS;

} // fim do programa
*/
```

```
//
// OBS.: RETIRAR OS COMENTARIOS /* */
//          PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
//

/*
// ------------------------------------ EXEMPLO601
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

void CONTAR (int X)
{
   If ( X > 0 )
   {
     CONTAR ( X-1 );
    cout << endl << X << endl;
 }
} // fim procedimento CONTAR( )

int main (void)
{
// PROGRAMA PARA CONTAR DE 1 ATE' 5, RECURSIVAMENTE

   cout << "EXEMPLO601 - CONTAR DE 1 A 5 RECURSIVAMENTE\n";
   CONTAR ( 5 );
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO602
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

void CONTAR (int X)
{
   If ( X > 0 )
   {
    cout << endl << X << endl;
    CONTAR ( X-1 );
   }
} // fim procedimento CONTAR( )

int main (void)
{
// PROGRAMA PARA CONTAR 5 10 ATE' 1, RECURSIVAMENTE

   cout << "EXEMPLO602 - CONTAR DE 5 A 1 RECURSIVAMENTE\n";
   CONTAR ( 5 );
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO603
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

void PARES (int X)
{
   if ( X > 0 )
     if ( X % 2 == 0 )
     {
       PARES ( X-2 );
       cout << endl << X << endl;
     }
     else
       PARES ( X-1 );
} // fim procedimento PARES ( )

int main (void)
{
// PROGRAMA RECURSIVO PARA MOSTRAR PARES

   cout << "EXEMPLO603 - MOSTRAR OS PARES <= 10\n";
   PARES ( 10 );
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO604
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

void PARES (int X)
{
   if ( X > 0 )
   {
     PARES ( X-1 );
     cout << endl << X << " " << 2*X << endl;
   }
} // fim procedimento PARES ( )

int main (void)
{
// PROGRAMA RECURSIVO PARA MOSTRAR PARES

   cout << "EXEMPLO604 - MOSTRAR OS 5 PRIMEIROS PARES\n";
   PARES ( 5 );
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO605
                                // bibliotecas de funcoes auxiliares
#include <iostream>      // para cin e cout
using namespace std;

void PARES (int X, int &S)
{
   if( X > 0 )
   {   PARES ( X-1, S );   S = S + 2*X;  }
   else
      S = 0;
} // fim procedimento PARES ( )

int main (void)
{
// PROGRAMA RECURSIVO PARA SOMAR PARES
// DADO:
   int SOMA;

   cout << "EXEMPLO605 - SOMAR OS 5 PRIMEIROS PARES\n";
   PARES ( 5, SOMA );
   cout << "\nSOMA DOS 5 PRIMEIROS PARES = " << SOMA << endl;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO606
                                // bibliotecas de funcoes auxiliares
#include <iostream>      // para cin e cout
using namespace std;

int PARES (int X)
{
// DADO:
   int S;
   if ( X > 0 )
     S = 2*X + PARES(X-1);
   else
     S = 0;
   return ( S );
} // fim funcao PARES ( )

int main (void)
{
// PROGRAMA RECURSIVO PARA SOMAR PARES
// DADO :
   int SOMA;

   cout << "EXEMPLO606 - SOMAR OS 5 PRIMEIROS PARES\n";
   SOMA  =  PARES ( 5 );
   cout << "\nSOMA DOS 5 PRIMEIROS PARES = " << SOMA << endl;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO607
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

int PARES (int X)
{
// DADO :
   int S;

   if ( X > 0 )
     if ( X % 2 == 0 )
       S = 1 + PARES ( X-2 );
     else
       S = PARES ( X-1 );
   else
     S = 0;

   return S;
} // fim funcao PARES ( )

int main (void)
{
// PROGRAMA RECURSIVO PARA CONTAR PARES

   cout << "EXEMPLO607 - CONTAR OS PARES <= 10\n";
   cout << "\nPARES <= 10 = " << PARES(10) << endl;
   cout << "\nFORNECER <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO608
                                // bibliotecas de funcoes auxiliares
#include <string.h>        // para strlen()
#include <iostream>        // para cin e cout

using namespace std;


typedef   char  STRING30[30];

bool PROCURAR
      (char LETRA, STRING30 S, int POSICAO)
{
// DADO :
   bool R;

   if( POSICAO <= int (strlen(S)) )
     R = (S[POSICAO] == LETRA) || PROCURAR(LETRA,S,POSICAO+1);
   else
     R = false;

   return ( R );
} // fim funcao PROCURAR ( )

int main (void)
{
// PROGRAMA RECURSIVO PARA ACHAR A POSICAO DE UMA LETRA
// DADO :
   char          L;
   STRING30 S;

   cout << "EXEMPLO608 - PROCURAR UMA LETRA EM UMA SENTENCA\n";
   cout << "\nFORNECER UMA SENTENCA COM MENOS DE 30 CARACTERES : ";
   cin.getline ( S, 30 );
   cout << "\nFORNECER UMA LETRA PARA SER PROCURADA : ";
   cin.get ( L );
   cout << "\nRESPOSTA = " << PROCURAR(L,S,0) << endl;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO609
                                // bibliotecas de funcoes auxiliares
#include <string.h>
#include <iostream>       // para cin e cout

using namespace std;


typedef   char  STRING30[30];

int PROCURAR
    (char LETRA, STRING30 S, int POSICAO)
{
// DADO :
   int R;

   if( POSICAO <= int (strlen(S)) )
     if( S[POSICAO] == LETRA )
       R = POSICAO + 1;  // a primeira posicao e' 0 !
     else
       R = PROCURAR(LETRA,S,POSICAO+1);
   else
     R = 0;

   return ( R );
} // fim funcao PROCURAR ( )

int main (void)
{
// PROGRAMA RECURSIVO PARA PROCURAR UMA LETRA
// DADO :
   char          L;
   STRING30 S;

   cout << "EXEMPLO609 - POSICAO DE UMA LETRA EM UMA SENTENCA\n";
   cout << "\nFORNECER UMA SENTENCA COM MENOS DE 30 CARACTERES : ";
   cin.getline ( S, 30 );
   cout << "\nFORNECER UMA LETRA PARA SER PROCURADA : ";
   cin.get ( L );
   cout << "\nRESPOSTA = " << PROCURAR(L,S,0) << endl;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                  // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO610
                                // bibliotecas de funcoes auxiliares
#include <string.h>
#include <iostream>       // para cin e cout

using namespace std;


typedef   char  STRING30[30];

int PROCURAR
   (char LETRA, STRING30 S, int POSICAO)
{
// DADO :
   int R;

   if( POSICAO <= int(strlen(S)) )
     if( S[POSICAO] == LETRA )
       R = 1 + PROCURAR(LETRA,S,POSICAO+1);
     else
       R = PROCURAR(LETRA,S,POSICAO+1);
   else
     R = 0;

   return ( R );
} // fim funcao PROCURAR ( )

int main (void)
{
// PROGRAMA RECURSIVO PARA PROCURAR OCORRENCIAS DE UMA LETRA
// DADO :
   char          L;
   STRING30 S;

   cout << "EXEMPLO610 - PROCURAR OCORRENCIAS DE UMA LETRA\n";
   cout << "\nFORNECER UMA SENTENCA COM MENOS DE 30 CARACTERES : ";
   cin.getline ( S, 30 );
   cout << "\nFORNECER UMA LETRA PARA SER PROCURADA : ";
   cin.get ( L );
   cout << "\nRESPOSTA = " << PROCURAR( L,S,0 ) << endl;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                  // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
//
// OBS.: RETIRAR OS COMENTARIOS /* */
//           PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
//

/*
// ------------------------------------ EXEMPLO701
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

typedef  int  TABELA [10];

int main ( void )
{
// PROGRAMA PARA LER UMA TABELA DE INTEIROS

// VARIAVEIS:
   TABELA V;
   int          X;

   cout << "EXEMPLO701 - LER UM TABELA DE 10 INTEIROS\n";
                                // REPETIR PARA CADA POSICAO
   for( X = 0; X < 10; X++  )
   {                            // a primeira posicao e' zero !
    cout << "\nFORNECER O " << (X+1) << "o. INTEIRO : ";
    cin  >>  V[ X ];
   }                            // FIM REPETIR
   cout << "\nVETOR LIDO: \n";
                                // REPETIR PARA CADA POSICAO
   for( X = 0; X < 10; X++  )
   {
     cout << V[ X ] << " ";
  }                             // FIM REPETIR

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO702
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

typedef  int  TABELA[10];

int main ( void )
{
// PROGRAMA PARA SOMAR UMA TABELA DE INTEIROS
// VARIAVEIS :
   TABELA V;
   int          X,
                SOMA;

   cout << "EXEMPLO702 - SOMAR UM TABELA DE 10 INTEIROS\n";

                                // REPETIR PARA CADA POSICAO
   for( X = 0; X < 10; X++  )
   {
     cout << "\nFORNECER O " << (X+1) << "o. INTEIRO : ";
     cin  >>  V[ X ];
   }                            // FIM REPETIR

   SOMA = 0;
                                // REPETIR PARA CADA POSICAO
   for( X = 0; X < 10; X++ )
     SOMA = SOMA + V[ X ];

   cout << "\nSOMA = " << SOMA;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO703
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

typedef  int  TABELA[10];

int main ( void )
{
// PROGRAMA PARA CALCULAR A MEDIA DE UMA TABELA DE INTEIROS
// VARIAVEIS :
   TABELA V;
   float  MEDIA;
   int     X,
           SOMA;

   cout << "EXEMPLO703 - MEDIA DE UMA TABELA DE 10 INTEIROS\n";

                                // REPETIR PARA CADA POSICAO
   for( X = 0; X < 10; X++  )
   {
     cout << "\nFORNECER O " << (X+1) << "o. INTEIRO : ";
     cin  >>  V[ X ];
   }                            // FIM REPETIR

   SOMA = 0;
                                // REPETIR PARA CADA POSICAO
   for( X = 0; X < 10; X++ )
     SOMA = SOMA + V[ X ];

   MEDIA = SOMA / 10.0;
   cout << "\nMEDIA = " << MEDIA;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO704
                                // bibliotecas de funcoes auxiliares
#include <string.h>        // para strlen()
#include <iostream>        // para cin e cout

using namespace std;

typedef char STRING10[10];

int main ( void )
{
// PROGRAMA PARA LER UMA PALAVRA
// VARIAVEIS :
   STRING10 PALAVRA;
   int             X;

   cout << "EXEMPLO704 - LER UMA PALAVRA\n";
   cout << "\nFORNECER UMA PALAVRA (NO MAXIMO 09 LETRAS) : ";
   cin  >>  PALAVRA;

   cout << "\nLETRAS DA PALAVRA LIDA : ";
                                // REPETIR PARA CADA POSICAO
   for( X = 0; X < strlen(PALAVRA); X++ )
     cout << PALAVRA[ X ] << " ";

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
/*
// ------------------------------------ EXEMPLO705
                                // bibliotecas de funcoes auxiliares
#include <string.h>        // para strlen()
#include <iostream>        // para cin e cout

using namespace std;

typedef char STRING10[10];

int main ( void )
{
// PROGRAMA PARA PROCURAR LETRA EM PALAVRA
// VARIAVEIS :
   char          LETRA;
   STRING10 PALAVRA;
   int             X;
   bool          ACHAR;

   cout << "EXEMPLO705 - PROCURAR LETRA EM UMA PALAVRA\n";
```

```
   cout << "\nDIGITAR UMA PALAVRA (NO MAXIMO 09 LETRAS) : ";
   cin   >>  PALAVRA;

   cout << "\nFORNECER A LETRA A SER PROCURADA : ";
   cin   >>  LETRA;

   ACHAR =  false;
   X          =  0;
                              // REPETIR PARA CADA POSICAO
   while( X < strlen(PALAVRA) && ! ACHAR )
   {
     if( PALAVRA[ X ] == LETRA )
       ACHAR = true;
    else
       X = X + 1;
   }                          // FIM REPETIR

   if( ACHAR )
     cout << "LETRA ENCONTRADA";
   else
     cout << "LETRA NAO ENCONTRADA";

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );               // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO706
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

typedef   float POLINOMIO[10+1];    // posicoes de 0:10

int main ( void )
{
// PROGRAMA PARA AVALIAR UM POLINOMIO
// VARIAVEIS :
   POLINOMIO P;
   int            Y, N;
   float        X, PX;

   cout << "EXEMPLO706 - LER COEFICIENTES DE UM POLINOMIO\n";

   cout << "\nFORNECER O GRAU DO POLINOMIO : ";
   cin   >>  N;

                                // REPETIR PARA CADA POSICAO
   for( Y = 0; Y <= N; Y++  )
   {
     cout << "\nFORNECER O " << (Y+1) << "o COEFICIENTE : ";
     cin   >>  P[ Y ];
   }                            // FIM REPETIR

   cout << "\nFORNECER O PONTO DE AVALIACAO : ";
   cin  >>  X;

   PX = 0;
                                // REPETIR PARA CADA POSICAO
                                // DA ULTIMA ATE' A PRIMEIRA
   for( Y = N; Y >= 0; Y-- )
     PX = PX * X + P[ Y ];

   cout << "\nP" << X << " = " << PX;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO707
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef   int VETOR[3]; // X,Y,Z

int main ( void )
{
// PROGRAMA PARA AVALIAR O COMPRIMENTO DE UM VETOR
// VARIAVEIS :
   VETOR V;
   inT         X;
   float       SOMA;

   cout << "EXEMPLO707 - COMPRIMENTO DE UM VETOR\n";

   cout << "\nFORNECER O VALOR DE X :"; cin >> V[0];
   cout << "\nFORNECER O VALOR DE Y :"; cin >> V[1];
   cout << "\nFORNECER O VALOR DE Z :"; cin >> V[2];

   SOMA = 0.0;
                                // REPETIR PARA CADA POSICAO
   for( X = 0; X < 3; X++ )
     SOMA = SOMA + V[ X ]*V[ X ];

   cout << "\nCOMPRIMENTO = " << sqrt(SOMA);

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO708
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;
typedef   int MATRIZ[2][2];

int main ( void )
{
// PROGRAMA PARA LER UMA MATRIZ
// VARIAVEIS :
 MATRIZ M;
   int        X, Y;

   cout << "EXEMPLO708 - LER UMA MATRIZ INTEIRA 2x2\n";

                                // REPETIR PARA CADA LINHA
   for ( X = 0; X < 2; X++ )
   {                            //  REPETIR PARA CADA COLUNA
     for ( Y = 0; Y < 2; Y++ )
     {
       cout << "\nFORNECER ELEMENTO " << (X+1) << "," << (Y+1) << " : ";
       cin  >>  M[ X ][ Y ];
     }                          //  FIM REPETIR
   }                            // FIM REPETIR

   cout <<  "\n";

                                // REPETIR PARA CADA LINHA
   for ( X = 0; X < 2; X++ )
   {                            //  REPETIR PARA CADA COLUNA
     for ( Y = 0; Y < 2; Y++ )
       cout << M[ X ][ Y ] << " ";
     cout <<  "\n";
   }                            // FIM REPETIR

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO709
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

typedef   int MATRIZ[2][2];

int main ( void )
{
// PROGRAMA PARA MONTAR A TRANSPOSTA DE UMA MATRIZ
// VARIAVEIS :
   MATRIZ M,              // MATRIZ ORIGINAL
          MT;             // MATRIZ TRANSPOSTA
   int    X, Y;

   cout << "EXEMPLO709 - TRANSPOR UMA MATRIZ INTEIRA 2x2\n";

                          // REPETIR PARA CADA LINHA
   for ( X = 0; X < 2; X++ )
   {                      //  REPETIR PARA CADA COLUNA
     for ( Y = 0; Y < 2; Y++ )
     {
       cout << "\nFORNECER ELEMENTO " << (X+1) << "," << (Y+1) << " : ";
       cin  >>  M[ X ][ Y ];
       MT[ Y ][ X ] = M[ X ][ Y ];
     }                    //  FIM REPETIR
   }                      // FIM REPETIR

   cout << "\n";
                          // REPETIR PARA CADA LINHA
   for ( X = 0; X < 2; X++ )
   {                      //  REPETIR PARA CADA COLUNA
     for ( Y = 0; Y < 2; Y++ )
       cout << MT[ X ][ Y ] << " ";
     cout <<  "\n";
   }                      // FIM REPETIR

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );           // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO710
                                // bibliotecas de funcoes auxiliares
#include <iostream>       // para cin e cout

using namespace std;

const      int ORDEM  = 3;

typedef   int MATRIZ [ ORDEM ][ ORDEM ];

int main ( void )
{
// PROGRAMA PARA MOSTRAR A DIAGONAL DE UMA MATRIZ
// VARIAVEIS :
   MATRIZ M;
   int          X, Y;

   cout << "EXEMPLO710 - MOSTRAR A DIAGONAL DE UMA MATRIZ\n";
                                // REPETIR PARA CADA LINHA
   for ( X = 0; X < ORDEM; X++ )
   {                            //  REPETIR PARA CADA COLUNA
     for ( Y = 0; Y < ORDEM; Y++ )
     {
       cout << "\nFORNECER ELEMENTO " << (X+1) << "," << (Y+1) << " : ";
       cin  >>  M[ X ][ Y ];
     }                          //  FIM REPETIR
   }                            // FIM REPETIR
   cout << "\n";
                                // REPETIR PARA CADA LINHA
   for ( X = 0; X < ORDEM; X++ )
   {                            //  REPETIR PARA CADA COLUNA
     for ( Y = 0; Y < ORDEM; Y++ )
     {
       if ( X == Y )            //   SE ESTIVER NA DIAGONAL
         cout << M[ X ][ Y ] << " ";
     }                          //  FIM REPETIR
   }                            // FIM REPETIR

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
//
// OBS.: RETIRAR OS COMENTARIOS /* */
//           PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
//

/*
// ------------------------------------ EXEMPLO801
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef
   struct SPONTOS
   {
     float X,Y,Z;
   } PONTOS;

int main ( void )
{
// PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
// DADOS:
   PONTOS P1, P2, P3;
   float       D;

   cout << "EXEMPLO801 - DISTANCIA ENTRE PONTOS\n";
   cout << "\n ENTRE COM O PRIMEIRO PONTO : \n";
   cin   >>  P1.X; cin >> P1.Y; cin >> P1.Z;
   cout << "\n ENTRE COM O SEGUNDO PONTO : \n";
   cin   >>  P2.X; cin >> P2.Y; cin >> P2.Z;
   P3.X = P2.X - P1.X;
   P3.Y = P2.Y - P1.Y;
   P3.Z = P2.Z  - P1.Z;
   D = sqrt ( pow(P3.X, 2.0)+
                pow(P3.Y, 2.0)+
                pow(P3.Z, 2.0) );
   cout << "\n DISTANCIA = " << D;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO802
                                // bibliotecas de funcoes auxiliares
#include   <math.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAM PARA CALCULAR A DISTANCIA ENTRE PONTOS
// DADOS:
   PONTOS P1, P2;
   float       D;

   cout << "EXEMPLO802 - DISTANCIA ENTRE PONTOS\n";
   cout << "ENTRE COM O PRIMEIRO PONTO : \n";
   cin   >>  P1.X; cin >> P1.Y; cin >> P1.Z;
   cout << "ENTRE COM O SEGUNDO  PONTO : \n";
   cin   >>  P2.X; cin >> P2.Y; cin >> P2.Z;
   D = sqrt ( pow(P2.X-P1.X, 2.0)+
                 pow(P2.Y-P1.Y, 2.0)+
                 pow(P2.Z-P1.Z, 2.0) );
   cout << "\n DISTANCIA = " << D;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                  // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO803
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;
typedef PONTOS VETOR[2];

int main ( void )
{
// PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
// DADOS:
 VETOR V;
   float D;

   cout << "EXEMPLO803 - DISTANCIA ENTRE PONTOS\n";
   cout << "\n ENTRE COM O PRIMEIRO PONTO : \n";
   cin   >>  V[0].X; cin >> V[0].Y; cin >> V[0].Z;
   cout << "\n ENTRE COM O SEGUNDO  PONTO : \n";
   cin   >>  V[1].X; cin >> V[1].Y; cin >> V[1].Z;
   D = sqrt ( pow(V[1].X-V[0].X, 2.0)+
                 pow(V[1].Y-V[0].Y, 2.0)+
                 pow(V[1].Z-V[0].Z, 2.0) );
   cout << "\n DISTANCIA = " << D;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO804
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef float PONTOS[3]; // X, Y, Z
typedef
   struct SVETOR
   { PONTOS P1, P2; } VETOR;

int main ( void )
{
// PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
// DADOS:
   VETOR V;
   float D;

   cout << "EXEMPLO804 - DISTANCIA ENTRE PONTOS\n";
   cout << "\nENTRE COM O PRIMEIRO PONTO : \n";
   cin   >>   V.P1[0]; cin >> V.P1[1]; cin >> V.P1[2];
   cout << "\nENTRE COM O SEGUNDO  PONTO : \n";
   cin   >>   V.P2[0]; cin >> V.P2[1]; cin >> V.P2[2];
   D = sqrt ( pow(V.P2[0]-V.P1[0], 2.0)+
                pow(V.P2[1]-V.P1[1], 2.0)+
                pow(V.P2[2]-V.P1[2], 2.0) );
   cout << "\n DISTANCIA = " << D;
   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO805
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef struct SPONTO  { float X,Y,Z; } PONTO ;
typedef struct SPONTOS { PONTO P1,P2; } PONTOS;

int main ( void )
{
// PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
// DADOS:
   PONTOS P;
   float       D;

   cout << "EXEMPLO805 - DISTANCIA ENTRE PONTOS\n";
   cout << "\n ENTRE COM O PRIMEIRO PONTO : \n";
   cin   >>  P.P1.X; cin >> P.P1.Y; cin >> P.P1.Z;
   cout << "\nENTRE COM O SEGUNDO  PONTO : \n";
   cin   >>  P.P2.X; cin >> P.P2.Y; cin >> P.P2.Z;
   D = sqrt ( pow(P.P2.X-P.P1.X, 2.0)+
                pow(P.P2.Y-P.P1.Y, 2.0)+
                pow(P.P2.Z-P.P1.Z,  2.0) );
   cout << "\n DISTANCIA = " << D;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO806
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef struct SPONTO  { float X,Y,Z; } PONTO ;
typedef float  VETOR[3];// X, Y, Z
typedef
   struct SPONTOS
   { PONTO P1;   VETOR P2; } PONTOS;

int main ( void )
{
// PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
// DADOS:
   PONTOS P;
   float         D;

   cout << "EXEMPLO806 - DISTANCIA ENTRE PONTOS\n";
   cout << "\n ENTRE COM O PRIMEIRO PONTO : \n";
   cin   >>  P.P1.X;  cin >> P.P1.Y;  cin >> P.P1.Z;
   cout << "\nENTRE COM O SEGUNDO  PONTO : \n";
   cin   >>  P.P2[0]; cin >> P.P2[1]; cin >> P.P2[2];
   D = sqrt ( pow(P.P2[0]-P.P1.X, 2.0)+
                pow(P.P2[1]-P.P1.Y, 2.0)+
                pow(P.P2[2]-P.P1.Z, 2.0));
   cout << "\n DISTANCIA = " << D;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO807
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef float VETOR[3]; // X, Y, Z
typedef VETOR PONTOS[2];

int main ( void )
{
// PROGRAMA PARA CALCULAR
// A DISTANCIA ENTRE PONTOS
// DADOS:
   PONTOS P;
   float         D;

   cout << "EXEMPLO807 - DISTANCIA ENTRE PONTOS\n";
   cout << "\n ENTRE COM O PRIMEIRO PONTO : \n";
   cin   >>  P[0][0]; cin >> P[0][1]; cin >> P[0][2];
   cout << "\n ENTRE COM O SEGUNDO  PONTO :";
   cin   >>  P[1][0]; cin >> P[1][1]; cin >> P[1][2];
   D = sqrt ( pow(P[1][0]-P[0][0], 2.0)+
                 pow(P[1][1]-P[0][1], 2.0)+
                 pow(P[1][2]-P[0][2], 2.0));
   cout << "\n DISTANCIA = " << D;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                  // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO808
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef float VETOR1[3];// X, Y, Z
typedef VETOR1 VETOR [2];

int main ( void )
{
// PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
// DADOS:
   VETOR P;
   float      D;

   cout << "EXEMPLO808 - DISTANCIA ENTRE PONTOS\n";
   cout << "ENTRE COM O PRIMEIRO PONTO : \n";
   cin   >>  P[0][0]; cin >> P[0][1]; cin >> P[0][2];
   cout << "ENTRE COM O SEGUNDO  PONTO : \n";
   cin   >>  P[1][0]; cin >> P[1][1]; cin >> P[1][2];
   D = sqrt( pow(P[1][0]-P[0][0], 2.0)+
                pow(P[1][1]-P[0][1], 2.0)+
                pow(P[1][2]-P[0][2], 2.0) );
   cout << "\n DISTANCIA = " << D;

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                  // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO809
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>          // para scanf ( ) e printf ( )

#include <iostream>        // para cin e cout

using namespace std;

typedef struct SPONTO  { float X,Y,Z; } PONTO ;
typedef PONTO  VETOR[2];// X, Y, Z

int main ( void )
{
// PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
// DADOS:
   VETOR P;
   float      D;

   cout << "EXEMPLO809 - DISTANCIA ENTRE PONTOS\n";
   cout << "\n ENTRE COM O PRIMEIRO PONTO : \n";
   scanf("%f%f%f", &(P[0].X), &(P[0].Y), &(P[0].Z));
   cout << "\n ENTRE COM O SEGUNDO  PONTO : \n";
   cin   >>  P[1].X; cin >> P[1].Y; cin >> P[1].Z;
   D = sqrt ( pow(P[1].X-P[0].X, 2.0)+
                 pow(P[1].Y-P[0].Y, 2.0)+
                 pow(P[1].Z-P[0].Z, 2.0) );
   printf( "\n DISTANCIA = %f", D);

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO810
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>          // para scanf ( ) e printf ( )

#include <iostream>        // para cin e cout

using namespace std;

typedef struct SPONTO  { float X,Y,Z; } PONTO ;
typedef struct SPONTOS { PONTO P1,P2; } PONTOS;

int main ( void )
{
// PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
// DADOS:
   PONTOS P;
  float          D;

   printf("EXEMPLO810 - DISTANCIA ENTRE PONTOS\n");
   printf("\n ENTRE COM O PRIMEIRO PONTO : \n");
   scanf("%f%f%f", &(P.P1.X), &(P.P1.Y), &(P.P1.Z));
   printf("\n ENTRE COM O SEGUNDO  PONTO : \n");
   scanf("%f%f%f", &(P.P2.X), &(P.P2.Y), &(P.P2.Z));
   D = sqrt ( pow(P.P2.X-P.P1.X, 2.0)+
                pow(P.P2.Y-P.P1.Y, 2.0)+
                pow(P.P2.Z-P.P1.Z, 2.0) );
   printf( "\n DISTANCIA = %f", D);

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
//
// OBS.: RETIRAR OS COMENTARIOS /* */
//           PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
//

/*
// ------------------------------------ EXEMPLO901
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <fstream>        // para arquivos
#include <iostream>       // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAMA PARA GRAVAR COORDENADAS DE PONTOS
// DADOS:
   PONTOS P;
   int           X;
   fstream    A;

   cout << "EXEMPLO901 - GRAVAR COORDENADAS DE PONTOS \n";
   A.open  ("PONTOS1.DAT", ios::out);
   for ( X = 1; X <= 2; X++ )
   {
     cout << "\nENTRE COM AS COORDENADAS DE UM PONTO : \n";
     cin   >>  P.X; cin >> P.Y; cin >> P.Z;
     A << P.X << "   " << P.Y <<  "  " << P.Z << endl;
   } // for
   A.close ( );

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO902
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <fstream>        // para arquivos
#include <iostream>       // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAMA PARA LER ARQUIVO DE PONTOS
// VARIAVEIS :
   PONTOS P;
   int           X;
   fstream    A;

   cout << "EXEMPLO902 - LER ARQUIVO DE PONTOS \n";
   A.open ("PONTOS1.DAT", ios::in);
   for ( X = 1; X <= 2; X++ )
   {
     A >> P.X >> P.Y >> P.Z;
     cout << "\nPONTO " << X << " : " << P.X << " " << P.Y << " " << P.Z;
   } // for
   A.close ( );

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO903
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <iostream>      // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAMA PARA GRAVAR COORDENADAS DE PONTOS
// DADOS:
   PONTOS P;
   int           X;
   FILE      *A;

   cout << "EXEMPLO903 - GRAVAR COORDENADAS DE PONTOS \n";
   A = fopen ( "PONTOS2.DAT", "wb" );
   for ( X = 1; X <= 2; X++ )
   {
     cout << "\nENTRE COM AS COORDENADAS DE UM PONTO : \n";
     cin  >>  P.X; cin >> P.Y; cin >> P.Z;
     fwrite ( &P, sizeof(PONTOS), 1, A );
   } // for
   fclose ( A );

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO904
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAMA PARA LER ARQUIVO DE PONTOS
// DADOS:
   PONTOS P;
   int            X;
   FILE       *A;

   cout << "EXEMPLO904 - LER ARQUIVO DE PONTOS \n";
   A = fopen ( "PONTOS2.DAT", "rb" );
   for( X = 1; X <= 2; X++ )
   {
     fread ( &P, sizeof(PONTOS), 1, A );
     printf ( "\nPONTO %d : %6.2f %6.2f %6.2f\n", X, P.X, P.Y, P.Z );
   } // for
   fclose(A);

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                  // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO905
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAMA PARA COPIAR O ARQUIVO COM COORDENADAS DE PONTOS
// DADOS:
   PONTOS P;
   int            X;
   FILE       *A1, *A2;

   cout << "EXEMPLO905 - COPIAR COORDENADAS DE PONTOS \n";
   A1 = fopen ( "PONTOS1.DAT", "rt"   );
   A2 = fopen ( "NOVO1.DAT"    , "wb" );
   for( X = 1; X <= 2; X++ )
   {
     fscanf ( A1, "%f%f%f\n", &P.X, &P.Y, &P.Z );
     fwrite  ( &P, sizeof(PONTOS), 1, A2 );
     printf  ( "\nCOPIADO %d : %6.2f %6.2f %6.2f\n", X, P.X, P.Y, P.Z );
   } // for
   fclose ( A1 );
   fclose ( A2 );

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                  // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO906
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <iostream>       // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAMA PARA ACRESCENTAR COORDENADAS DE PONTOS
// DADOS:
   PONTOS P;
   int           X;
   FILE       *A;

   cout << "EXEMPLO906 - ACRESCENTAR COORDENADAS DE PONTOS \n";
   A = fopen ( "PONTOS2.DAT", "r+b" );
   while( ! feof(A) )
     fread ( &P, sizeof(PONTOS), 1, A );   // LER ATE' O FIM DE ARQUIVO
   fseek ( A, 0L, SEEK_CUR );              // MARCAR A POSICAO
   for ( X = 1; X <= 2; X++ )
   {
     cout << "\nENTRE COM AS COORDENADAS DE OUTRO PONTO : \n";
     cin  >> P.X; cin >> P.Y; cin >> P.Z;
     fwrite ( &P, sizeof(PONTOS), 1, A );
   } // for
   fclose(A);

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO907
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <iostream>      // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAMA PARA ACRESCENTAR COORDENADAS DE PONTOS
// DADOS:
   PONTOS P;
   int           X;
   FILE      *A;

   cout << "EXEMPLO907 - ACRESCENTAR COORDENADAS DE PONTOS \n";
   A = fopen ( "PONTOS1.DAT", "ab" );
   for( X = 1; X <= 2; X++ )
   {
     cout << "\nENTRE COM AS COORDENADAS DE OUTRO PONTO : \n";
     cin  >>  P.X; cin >> P.Y; cin >> P.Z;
     fprintf ( A, "%f %f %f\n", P.X, P.Y, P.Z );
   } // for
   fclose( A );

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO908
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <bool.h>
#include <iostream>      // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAMA PARA PROCURAR COORDENADAS DE PONTOS
// DADOS:
   PONTOS  P,
           PROCURADO;
   bool    ACHAR;
   FILE   *A;

   cout << "EXEMPLO908 - PROCURAR COORDENADAS DE PONTOS \n";
   cout << "\nENTRE COM AS COORDENADAS DO PONTO A PROCURAR : \n";
   cin   >>  PROCURADO.X; cin >> PROCURADO.Y; cin >> PROCURADO.Z;
   ACHAR = false;
   A = fopen ( "PONTOS1.DAT", "r" );
   while( ! feof(A) && ! ACHAR)
   {
     fscanf ( A , "%f%f%f", &P.X, &P.Y, &P.Z );
     if (P.X==PROCURADO.X && P.Y==PROCURADO.Y && P.Z==PROCURADO.Z )
       ACHAR = true;
   } // while
   if( ACHAR )
     cout << "\nCOORDENADAS ENCONTRADAS";
   else
     cout << "\nCOORDENADAS NAO ENCONTRADAS";
   fclose ( A );

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO909
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <iostream>      // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAMA PARA ALTERAR COORDENADAS DE PONTOS
// DADOS:
   PONTOS P;
   int           X;
   FILE      *A;

   cout << "EXEMPLO909 - ALTERAR COORDENADAS DE PONTOS \n";
   A = fopen("PONTOS2.DAT", "r+b");
   for ( X = 1; X <= 2; X++ )
   {
     fseek ( A, (X-1)*sizeof(PONTOS), SEEK_SET ); // o primeiro e' zero
     fread ( &P, sizeof(PONTOS), 1, A );
     printf ( "\nATUAL %d : %6.2f %6.2f %6.2f\n", X, P.X, P.Y, P.Z );
     cout << "\nENTRE COM AS NOVAS COORDENADAS DO PONTO : \n";
     cin   >>  P.X; cin >> P.Y; cin >> P.Z;
     fseek ( A, (X-1)*sizeof(PONTOS), SEEK_SET );
     fwrite ( &P, sizeof(PONTOS), 1, A );
   } // for
   fclose(A);

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```

```
/*
// ------------------------------------ EXEMPLO910
                                // bibliotecas de funcoes auxiliares
#include <math.h>
#include <stdio.h>
#include <fcntl.h>
#include <iostream>      // para cin e cout

using namespace std;

typedef struct SPONTOS { float X,Y,Z; } PONTOS;

int main ( void )
{
// PROGRAMA PARA LER ARQUIVO DE PONTOS DIRETAMENTE
// DADOS:
   PONTOS P;
   int           X;
   FILE      *A;

   cout << "EXEMPLO910 - LER ARQUIVO DE PONTOS DIRETAMENTE \n";
   A = fopen ( "PONTOS2.DAT", "r" );
   for ( X = 2; X > 0; X-- )
   {
     fseek ( A, (X-1) * sizeof (PONTOS), SEEK_SET ); // o primeiro e' zero
     fread ( &P, sizeof(PONTOS), 1, A );
     printf ( "\nPONTO %d : %6.2f %6.2f %6.2f\n", X, P.X, P.Y, P.Z );
   } // for
   fclose( A );

   cout << "\nPRESSIONAR <Enter> PARA TERMINAR";
   getchar ( );                 // para esperar
   return EXIT_SUCCESS;
} // fim do programa
*/
```