

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

/**/
//
// ----- EXEMPLO101
//

class CS0101
{
// Documentacao automatica:
/// <summary>
/// Acao principal responsavel por acionar outros metodos.
/// </summary>
/// <returns>
/// Nao ha' retorno (void).
/// </returns>
/// <param name="args"> Lista de parametros
/// com dados da linha de comando.
/// </param>
/// <remarks>
/// A lista de parametros e' opcional.
/// </remarks>

    public static void Main ( string [ ] args ) {
        System.Console.WriteLine ( "EXEMPLO101 - PRIMEIRO EXEMPLO EM C#" );
        System.Console.WriteLine ( );
        System.Console.Write ( "\nApertar ENTER para terminar." );
        System.Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
/**/
/*
//
// ----- EXEMPLO102
//
// metodos auxiliares
using System;

class CS0102
{
    public static void Main ( )
    {
        Console.WriteLine ( "EXEMPLO102 - PRIMEIRO EXEMPLO EM C#" );
        Console.WriteLine ( );
        Console.Write ( "\nApertar ENTER para continuar." );
        Console.ReadLine ( );
        Console.Clear ( );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
```

```

/*
//
// ----- EXEMPLO103
//
// metodos auxiliares
using System;

class CS0103
{
    public static void Main ( )
    {
        Console.WriteLine ( "EXEMPLO103 - PRIMEIRO EXEMPLO EM C#" );
        Console.WriteLine ( );
        Console.WriteLine ( "MATRICULA: _____ ALUNO : _____ " );
        Console.Write ( "\n" ); // para mudar de linha
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO104
//
// metodos auxiliares
using System;

class CS0104
{
    // definicao de constante do tipo caractere
    public const char endl = "\n";

    public static void Main ( )
    {
        Console.WriteLine ( "EXEMPLO104 - PRIMEIRO EXEMPLO EM C#" );
        Console.WriteLine ( );
        Console.WriteLine ( "MATRICULA: _____ ALUNO : _____ " );
        Console.Write ( endl ); // para mudar de linha
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO105
//
// metodos auxiliares
using System;

class CS0105
{
    // definicao de constante do tipo caractere
    public const char endl = '\n';

    public static void Main ( )
    {
        Console.WriteLine ( "EXEMPLO105 - PRIMEIRO EXEMPLO EM C#" );
        Console.WriteLine ( );
        Console.WriteLine ( "MATRICULA: _____ ALUNO : _____" );
        Console.Write ( endl ); // para mudar de linha
        Console.WriteLine ( "\nEXEMPLOS DE VALORES : " );
        Console.WriteLine ( "\nCARACTERE : " + 'A' ); // letra ou simbolo
        Console.WriteLine ( "\nINTEIRO : " + 10 ); // valor sem parte fracionaria
        Console.WriteLine ( "\nREAL : " + 3.14115 ); // valor com parte fracionaria
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO106
//
// metodos auxiliares
using System;

class CS0106
{
    // definicao de constante do tipo caractere
    public const char endl = '\n';
    // definicao de constante do tipo real
    public const double PI = 3.1415;

    public static void Main ( )
    {
        Console.WriteLine ( "EXEMPLO106 - PRIMEIRO EXEMPLO EM C#" );
        Console.WriteLine ( );
        Console.WriteLine ( "MATRICULA: _____ ALUNO : _____" );
        Console.Write ( endl ); // para mudar de linha
        Console.WriteLine ( "\nEXEMPLOS DE VALORES : " );
        Console.WriteLine ( "\nCARACTERE : " + 'A' ); // letra ou simbolo
        Console.WriteLine ( "\nINTEIRO : " + 10 ); // valor sem parte fracionaria
        Console.WriteLine ( "\nREAL : " + PI ); // valor com parte fracionaria
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO107
//
// metodos auxiliares
using System;

class CS0107
{
    // definicao de constante do tipo caractere
    public const char endl = '\n';
    // definicao de constante do tipo real
    public const double PI = 3.1415;

    public static void Main ( )
    {

        Console.WriteLine ( "EXEMPLO107 - PRIMEIRO EXEMPLO EM C#" );
        Console.WriteLine ( );
        Console.WriteLine ( "MATRICULA: _____ ALUNO : _____" );
        Console.Write ( endl ); // para mudar de linha
        Console.WriteLine ( "\nEXEMPLOS DE VALORES : " ); // saidas formatadas
        Console.WriteLine ( "\nCARACTERE : {0,6}" , 'A' ); // letra ou simbolo
        Console.WriteLine ( "\nINTEIRO : {0,6:D3}", 10 ); // valor sem parte fracionaria
        Console.WriteLine ( "\nREAL : {0,6:F3}", PI ); // valor com parte fracionaria

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO108
//
// metodos auxiliares
using System;

class CS0108
{
    // definicao de constante do tipo caractere
    public const char endl = '\n';
    // definicao de constante do tipo real
    public const double PI = 3.1415;

    public static void Main ( )
    {
        // definicao de variavel real
        double X = 10.01;                                // com atribuicao de valor inicial

        Console.WriteLine ( "EXEMPLO108 - PRIMEIRO EXEMPLO EM C#" );
        Console.WriteLine ( );
        Console.WriteLine ( "MATRICULA: _____ ALUNO : _____" );
        Console.Write ( endl );                            // para mudar de linha
        Console.WriteLine ( "\nEXEMPLOS DE VALORES : " );
        Console.WriteLine ( "\nCARACTERE : " + 'A' );     // letra ou simbolo
        Console.WriteLine ( "\nINTEIRO : " + 10 );        // valor sem parte fracionaria
        Console.WriteLine ( "\nREAL : " + PI );           // valor com parte fracionaria
        Console.WriteLine ( "\nREAL : " + X );            // valor com parte fracionaria
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO109
//
// metodos auxiliares
using System;

class CS0109
{
    // definicao de constante do tipo caractere
    public const char endl = '\n';
    // definicao de constante do tipo real
    public const double PI = 3.1415;

    public static void Main ( )
    {
        // definicao de variavel real
        double X = 10.01;                                // com atribuicao de valor inicial
        // definicao de variavel inteira
        int I = 10;

        Console.WriteLine ( "EXEMPLO109 - PRIMEIRO EXEMPLO EM C#" );
        Console.WriteLine ( );
        Console.WriteLine ( "MATRICULA: _____ ALUNO : _____" );
        Console.Write ( endl );                            // para mudar de linha
        Console.WriteLine ( "\nEXEMPLOS DE VALORES : " );
        Console.WriteLine ( "\nCARACTERE : " + 'A' );      // letra ou simbolo
        Console.WriteLine ( "\nINTEIRO : " + 10 );         // valor sem parte fracionaria
        Console.WriteLine ( "\nINTEIRO : " + I );          // valor sem parte fracionaria
        Console.WriteLine ( "\nREAL : " + PI );            // valor com parte fracionaria
        Console.WriteLine ( "\nREAL : " + X );             // valor com parte fracionaria
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO110
//
// metodos auxiliares
using System;

class CS0110
{
    // definicao de constante do tipo caractere
    public const char endl = '\n';
    // definicao de constante do tipo real
    public const double PI = 3.1415;

    public static void Main ( )
    {
        // definicao de variavel real
        double X = 10.01;                // com atribuicao de valor inicial
        // definicao de variavel inteira
        int I = 10;
        // definicao de variavel caractere
        char N = '\n';                  // mudar de linha

        Console.WriteLine ( "EXEMPLO110 - PRIMEIRO EXEMPLO EM C#" );
        Console.WriteLine ( );
        Console.WriteLine ( "MATRICULA: _____ ALUNO : _____" );
        Console.Write ( endl );          // para mudar de linha
        Console.WriteLine ( N + "EXEMPLOS DE VALORES : " );
        Console.Write ( N + "CARACTERE : " + 'A' ); // letra ou simbolo
        Console.Write ( N + "INTEIRO : " + 10 );    // valor sem parte fracionaria
        Console.Write ( N + "INTEIRO : " + I );     // valor sem parte fracionaria
        Console.Write ( N + "REAL : " + PI );       // valor com parte fracionaria
        Console.Write ( N + "REAL : " + X );        // valor com parte fracionaria
        Console.WriteLine ( N + "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

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

/**/
//
// ----- EXEMPLO201
//
// metodos auxiliares
using System;

class CS0201
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E MOSTRAR UM VALOR INTEIRO
        // VARIABEL :
        int X;

        Console.WriteLine ( "EXEMPLO201 - LER E MOSTRAR UM VALOR INTEIRO" );
        Console.Write ( "\nFORNECER UM VALOR INTEIRO QUALQUER : " );
        X = int.Parse ( Console.ReadLine ( ) );
        Console.WriteLine ( "\nO VALOR DIGITADO FOI : " + X );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
/**/
/*
//
// ----- EXEMPLO202
//
// metodos auxiliares
using System;

class CS0202
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E MOSTRAR UM VALOR REAL
        // VARIABEL :
        double X;

        Console.WriteLine ( "EXEMPLO202 - LER E MOSTRAR UM VALOR REAL" );
        Console.Write ( "\nFORNECER UM VALOR REAL QUALQUER : " );
        X = double.Parse ( Console.ReadLine ( ) );
        Console.WriteLine ( "\nO VALOR DIGITADO FOI : " + X );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
```



```

*/
/*
//
// ----- EXEMPLO203
//
// metodos auxiliares
using System;

class CS0203
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E MOSTRAR UM CARACTERE
        // VARIABEL :
        char X;

        Console.WriteLine ( "EXEMPLO203 - LER E MOSTRAR UM CARACTERE" );
        Console.Write      ( "\nFORNECER UM CARACTERE QUALQUER : " );
        X = (char) Console.Read ( );
        Console.ReadLine ( );    // retirar a mudanca de linha pendente (limpar o buffer)
        Console.WriteLine ( "\nO CARACTERE DIGITADO FOI : " + X );
        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO204
//
// metodos auxiliares
using System;

class CS0204
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E MOSTRAR CARACTERES
        // VARIABEL :
        string X;

        Console.WriteLine ( "EXEMPLO204 - LER E MOSTRAR CARACTERES" );
        Console.Write      ( "\nFORNECER UMA CADEIA DE CARACTERES QUALQUER : " );
        X = Console.ReadLine ( );
        Console.WriteLine ( "\nA CADEIA DE CARACTERES DIGITADA FOI : " + X );
        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO205
//
// metodos auxiliares
using System;

class CS0205
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E SOMAR DOIS VALORES INTEIROS
        // VARIAVEIS :
        int X,Y,Z;

        Console.WriteLine ( "EXEMPLO205 - LER E SOMAR DOIS VALORES INTEIROS" );
        Console.Write ( "\nFORNECER UM VALOR INTEIRO QUALQUER : " );
        X = Convert.ToInt32 ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER OUTRO VALOR INTEIRO QUALQUER : " );
        Y = Convert.ToInt32 ( Console.ReadLine ( ) );
        Z = X + Y;
        Console.WriteLine ( "\nA SOMA DOS DOIS = " + Z );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO206
//
// metodos auxiliares
using System;

class CS0206
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E SUBTRAIR DOIS VALORES REAIS
        // VARIAVEIS :
        double X,Y,Z;

        Console.WriteLine ( "EXEMPLO206 - LER E SOMAR DOIS VALORES INTEIROS" );
        Console.Write ( "\nFORNECER UM VALOR REAL QUALQUER : " );
        X = Convert.ToDouble( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER OUTRO VALOR REAL QUALQUER : " );
        Y = Convert.ToDouble( Console.ReadLine ( ) );
        Z = X - Y;
        Console.WriteLine ( "\nA DIFERENCA ENTRE DOS DOIS = " + Z );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO207
//
// metodos auxiliares
using System;

class CS0207
{
    public static void Main ( )
    {
        // PROGRAMA PARA OPERAR VALORES LOGICOS
        // VARIAVEIS :
        bool X,Y,Z;

        Console.WriteLine ( "EXEMPLO207 - OPERAR VALORES LOGICOS" );
        X = true;
        Y = false;
        Z = X || Y;      // X ou Y
        Console.WriteLine ( "\nA DISJUNCAO ENTRE VERDADEIRO E FALSO = " + Z );
        Z = X && Y;      // X e Y
        Console.WriteLine ( "\nA CONJUNCAO ENTRE VERDADEIRO E FALSO = " + Z );
        Z = ! Z;        // nao Z
        Console.WriteLine ( "\nA NEGACAO DO VALOR ANTERIOR E' IGUAL A " + Z );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO208
//
// metodos auxiliares
using System;

class CS0208
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A VELOCIDADE DE UM VEICULO
        // VARIAVEIS :
        double D,      // Distancia
               T,      // Tempo
               V;       // Velocidade

        Console.WriteLine ( "EXEMPLO208 - CALCULAR A VELOCIDADE DE UM VEICULO" );
        Console.Write ( "\nFORNECER UMA DISTANCIA QUALQUER EM METROS: " );
        D = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O TEMPO PARA PERCORRE-LA EM SEGUNDOS: " );
        T = double.Parse ( Console.ReadLine ( ) );
        V = D / T;
        Console.WriteLine ( "\nV = D / T = " + V + " m/s " );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO209
//
// metodos auxiliares
using System;

class CS0209
{
    public static void Main ( )
    {
        // PROGRAMA PARA COMPARAR CARACTERES COM UMA SENHA
        // CONSTANTE :
        const string SENHA = "XXXX";
        // VARIABEL :
        string S;

        Console.WriteLine ( "EXEMPLO209 - COMPARAR CARACTERES COM UMA SENHA" );
        Console.Write ( "\nFORNECER UMA CADEIA DE CARACTERES QUALQUER: " );
        S = Console.ReadLine ( );
        Console.WriteLine ( "\nA COMPARACAO COM A SENHA = " +
            ((String.Compare(S,SENHA)==0)?true:false) );
        // String.Compare(S1,S2) compara as cadeias S1 com S2
        // igual a 0 : S1 = S2 => 1 ( verdadeiro )
        // diferente : S1 <> S2 => 0 ( falso )
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO210
//
// metodos auxiliares
using System;

class CS0210
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR O ARCO TRIGONOMETRICO DE UM SENO
        // CONSTANTE :
        const double PI = 3.14;
        // VARIAVEIS :
        double ARCO, COSSENO, SENO, TANGENTE;

        Console.WriteLine ( "EXEMPLO210 - CALCULAR O ARCO TRIGONOMETRICO DE UM SENO" );
        Console.Write ( "\nFORNECER O VALOR DO SENO : " );
        SENO = double.Parse ( Console.ReadLine ( ) );
        COSSENO = Math.Sqrt( 1.0 - Math.Pow(SEN0,2));
        TANGENTE = SENO / COSSENO;
        ARCO = Math.Atan(TANGENTE);
        Console.WriteLine ( "\nO ARCO TRIGONOMETRICO EM GRAUS = " + (ARCO*180.0/PI) );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

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

/**/
//
// ----- EXEMPLO301
//
// metodos auxiliares
using System;

class CS0301
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER UM VALOR INTEIRO E VERIFICAR SE E' ZERO
        // VARIABEL:
        int X;

        Console.WriteLine ( "EXEMPLO301 - LER E TESTAR UM VALOR INTEIRO" );
        Console.Write ( "\nFORNECER UM VALOR INTEIRO QUALQUER: " );
        X = int.Parse ( Console.ReadLine ( ) );
        if( X == 0 )
            Console.WriteLine ( "\nO VALOR DIGITADO FOI ZERO" );
        else
            Console.WriteLine ( "\nO VALOR DIGITADO NAO FOI ZERO" );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
/**/
/*
//
// ----- EXEMPLO302
//
// metodos auxiliares
using System;

class CS0302
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER UM REAL E TESTAR SE DIFERENTE DE ZERO
        // VARIABEL:
        double X;

        Console.WriteLine ( "EXEMPLO302 - LER E TESTAR UM VALOR REAL" );
        Console.Write ( "\nFORNECER UM VALOR REAL DIFERENTE DE ZERO: " );
        X = double.Parse ( Console.ReadLine ( ) );
        if( X != 0.0 )
            Console.WriteLine ( "\nO VALOR DIGITADO FOI DIFERENTE DE ZERO" );
        else
            Console.WriteLine ( "\nO VALOR DIGITADO NAO FOI DIFERENTE DE ZERO" );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
```

```

/*
//
// ----- EXEMPLO303
//
// metodos auxiliares
using System;

class CS0303
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER CARACTERE E VERIFICAR SE E' UM ALGARISMO
        // VARIABEL:
        char X;

        Console.WriteLine ( "EXEMPLO303 - LER E TESTAR CARACTERE" );
        Console.Write ( "\nFORNECER UM ALGARISMO QUALQUER: " );
        X = (char) Console.Read ( );
        Console.ReadLine ( ); // retirar a mudanca de linha pendente (limpar o buffer)
        if( X >= '0' && X <= '9' )
        {
            Console.WriteLine ( "\nO VALOR DIGITADO FOI UM ALGARISMO" );
            Console.WriteLine ( "\nO ALGARISMO DIGITADO FOI : " + X );
        } // if ALGARISMO
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO304
//
// metodos auxiliares
using System;

class CS0304
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER CARACTERE E TESTAR SE NAO E' UM ALGARISMO
        // VARIABEL:
        char X;

        Console.WriteLine ( "EXEMPLO304 - LER E TESTAR CARACTERE" );
        Console.Write ( "\nFORNECER UM CARACTERE QUALQUER: " );
        X = (char) Console.Read ( );
        Console.ReadLine ( ); // retirar a mudanca de linha pendente (limpar o buffer)
        if( !( X >= '0' && X <= '9' ) )
        {
            Console.WriteLine ( "\nNAO FOI DIGITADO UM ALGARISMO" );
            Console.WriteLine ( "\nFOI DIGITADO O CARACTERE: " + X );
        } // if NAO ALGARISMO
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO305
//
// metodos auxiliares
using System;

class CS0305
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E TESTAR A IGUALDADE DE DOIS INTEIROS
        // VARIAVEIS:
        int X,Y;

        Console.WriteLine ( "EXEMPLO305 - LER E TESTAR DOIS VALORES INTEIROS" );
        Console.Write      ( "\nFORNECER UM VALOR INTEIRO QUALQUER: " );
        X = int.Parse      ( Console.ReadLine ( ) );
        Console.Write      ( "\nFORNECER OUTRO VALOR INTEIRO QUALQUER: " );
        Y = int.Parse      ( Console.ReadLine ( ) );
        if( X == Y )
            Console.WriteLine ( "\nDOIS VALORES IGUAIS" );
        else
            Console.WriteLine ( "\n" + X + " DIFERENTE DE " + Y );
        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO306
//
// metodos auxiliares
using System;

class CS0306
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E TESTAR DOIS VALORES REAIS
        // VARIAVEIS:
        double X,Y;

        Console.WriteLine ( "EXEMPLO306 - LER E TESTAR DOIS VALORES REAIS" );
        Console.Write   ( "\nFORNECER UM VALOR REAL QUALQUER: " );
        X = double.Parse ( Console.ReadLine ( ) );
        Console.Write   ( "\nFORNECER OUTRO VALOR REAL QUALQUER: " );
        Y = double.Parse ( Console.ReadLine ( ) );
        if( !(X == Y) )
        {
            Console.Write ( "\n" + X );
            Console.Write ( " DIFERENTE DE " );
            Console.Write ( Y );
        }
        else
        {
            Console.Write ( "\nVALORES IGUAIS" );
        } // if VALORES DIFERENTES
        Console.WriteLine ( );
        Console.Write   ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```



```

/*
//
// ----- EXEMPLO307
//
// metodos auxiliares
using System;

class CS0307
{
    public static void Main ( )
    {
        // PROGRAMA PARA TRATAR ALTERNATIVAS COM VALORES LOGICOS
        // VARIAVEIS:
        int X,Y;
        bool Z;

        Console.WriteLine ( "EXEMPLO307 - TRATAR VALORES LOGICOS" );
        Console.Write ( "\nFORNECER UM VALOR INTEIRO QUALQUER: " );
        Z = Int32.TryParse( Console.ReadLine ( ), out X );
        Console.Write ( "\nFORNECER OUTRO VALOR INTEIRO QUALQUER: " );
        Z = Z && Int32.TryParse( Console.ReadLine ( ), out Y );
        if ( ! Z )
        {
            Console.WriteLine( "\nERRO: ENTRADA DE DADOS INCORRETA" );
        }
        else
        {
            Z = ( X == Y );
            if( Z )
                Console.WriteLine( "\nVALORES IGUAIS" );
            else
                Console.WriteLine( "\nVALORES DIFERENTES" );
        } // fim se
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO308
//
// metodos auxiliares
using System;

class CS0308
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER CARACTERE E TESTAR UMA LETRA
        // VARIÁVEL:
        char X;

        Console.WriteLine ( "EXEMPLO308 - LER E TESTAR UMA LETRA" );
        Console.Write      ( "\nFORNECER UMA LETRA QUALQUER: " );
        X = (char) Console.Read ( );
        Console.ReadLine ( );    // retirar a mudança de linha pendente (limpar o buffer)
        if( X >= 'A' && X <= 'Z' )
            Console.WriteLine ( "\nFOI DIGITADA UMA LETRA MAIUSCULA" );
        else
            if( X >= 'a' && X <= 'z' )
                Console.WriteLine ( "\nFOI DIGITADA UMA LETRA MINUSCULA" );
            else
                Console.WriteLine ( "\nNAO FOI DIGITADA UMA LETRA" );
        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da ação principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO309
//
// metodos auxiliares
using System;

class CS0309
{
    public static void Main ( )
    {
        // PROGRAMA PARA COMPARAR CARACTERES < , = , >
        // CONSTANTES:
        const char MAIOR = '>',
                IGUAL = '=',
                MENOR = '<';
        // VARIABEL:
        char X;

        Console.WriteLine ( "EXEMPLO309 - COMPARAR CARACTERES < , = , >" );
        Console.Write      ( "\nFORNECER UM DOS CARACTERES CITADOS: " );
        X = (char) Console.Read ( );
        Console.ReadLine ( );    // retirar a mudanca de linha pendente (limpar o buffer)
        switch( X )
        {
            case MAIOR: Console.WriteLine ( "\nFOI DIGITADO O SINAL DE MAIOR" );
                        break;
            case IGUAL: Console.WriteLine ( "\nFOI DIGITADO O SINAL DE IGUAL" );
                        break;
            case MENOR: Console.WriteLine ( "\nFOI DIGITADO O SINAL DE MENOR" );
                        break;
            default      : Console.WriteLine ( "\nFOI DIGITADO UM OUTRO CARACTERE QUALQUER" );
                        break;
        } // COMPARACAO DE X COM < , = , >
        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO310
//
// metodos auxiliares
using System;

class CS0310
{
    public static void Main ( )
    {
        // PROGRAMA PARA IDENTIFICAR CARACTERES
        // VARIABEL:
        char X;

        Console.WriteLine ( "EXEMPLO310 - IDENTIFICAR CARACTERES" );
        Console.Write      ( "\nFORNECER UM CARACTERE QUALQUER: " );
        X = (char) Console.Read ( );
        Console.ReadLine ( );    // retirar a mudanca de linha pendente (limpar o buffer)
        switch ( X )
        {
            case 'A':
            case 'E':
            case 'I':
            case 'O':
            case 'U': Console.WriteLine ( "\nFOI DIGITADO UMA VOGAL" );
                       break;

            case '0':
            case '1':
            case '2':
            case '3':
            case '4':
            case '5':
            case '6':
            case '7':
            case '8':
            case '9': Console.WriteLine ( "\nFOI DIGITADO UM ALGARISMO" );
                       Console.WriteLine ( "\nO NUMERO CORRESPONDENTE = " + (X-48) );
                       break;

            default : Console.WriteLine ( "\nFOI DIGITADO UM OUTRO CARACTERE QUALQUER" );
                       break;
        } // IDENTIFICACAO DE UM CARACTERE
        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

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

/**/
//
// ----- EXEMPLO401
//
// metodos auxiliares
using System;

class CS0401
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E IMPRIMIR 03 VALORES INTEIROS
        // VARIAVEIS:
        int X,
            CONTADOR;

        Console.WriteLine ( "EXEMPLO401 - LER E IMPRIMIR 03 VALORES INTEIROS" );
        Console.WriteLine ( );           // mudar de linha
        CONTADOR = 1;
        while( CONTADOR <= 3 ) // REPETIR
        {
            Console.WriteLine ( );       // mudar de linha
            Console.Write      ( CONTADOR + " FORNECER UM VALOR INTEIRO: " );
            X = int.Parse      ( Console.ReadLine ( ) );
            Console.WriteLine ( "\nO VALOR DIGITADO FOI " + X );
            CONTADOR = CONTADOR + 1;
        }                               // ENQUANTO CONTADOR <= 3

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
/**/

```

```

/*
//
// ----- EXEMPLO402
//
// metodos auxiliares
using System;

class CS0402
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
        // VARIAVEIS:
        int X, N, CONTADOR;

        Console.WriteLine ( "EXEMPLO402 - LER E IMPRIMIR (N) VALORES INTEIROS\n" );
        Console.Write      ( "\nFORNECER O NUMERO DE VEZES (N): " );
        N = int.Parse      ( Console.ReadLine ( ) );
        CONTADOR = 1;
        while( CONTADOR <= N ) // REPETIR
        {
            Console.Write    ( "\n" + CONTADOR );
            Console.WriteLine ( " FORNECER UM VALOR INTEIRO QUALQUER: " );
            X = int.Parse    ( Console.ReadLine ( ) );
            Console.WriteLine ( "\nO VALOR DIGITADO FOI " + X );
            CONTADOR = CONTADOR + 1;
        }
            // ENQUANTO CONTADOR <= N

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO403
//
// metodos auxiliares
using System;

class CS0403
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
        // VARIAVEIS:
        int X, N;

        Console.WriteLine ( "EXEMPLO403 - LER E IMPRIMIR (N) VALORES INTEIROS\n" );
        Console.Write ( "\nFORNECER O NUMERO DE VEZES (N): " );
        N = int.Parse ( Console.ReadLine ( ) );
        while( N > 0 )           // REPETIR
        {
            Console.Write ( "\n" + N );
            Console.Write ( " FORNECER UM VALOR INTEIRO QUALQUER: " );
            X = int.Parse ( Console.ReadLine ( ) );
            Console.WriteLine ( "\nO VALOR DIGITADO FOI " + X );
            N = N - 1;
        }
        // ENQUANTO N > 0
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO404
//
// metodos auxiliares
using System;

class CS0404
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E IMPRIMIR 03 VALORES INTEIROS
        // VARIAVEIS:
        int X,
            CONTADOR;

        Console.WriteLine ( "EXEMPLO404 - LER E IMPRIMIR 03 VALORES INTEIROS\n" );
        for( CONTADOR = 1; CONTADOR <= 3; CONTADOR = CONTADOR+1 )
        {
            Console.Write ( "\n" + CONTADOR + " FORNECER UM VALOR INTEIRO: " );
            X = int.Parse ( Console.ReadLine ( ) );
            Console.WriteLine ( "\nO VALOR DIGITADO FOI " + X );
        } // PARA CONTADOR EM [1:3]

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO405
//
// metodos auxiliares
using System;

class CS0405
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
        // VARIAVEIS:
        int X, N,
            CONTADOR;

        Console.WriteLine ( "EXEMPLO405 - LER E IMPRIMIR (N) VALORES INTEIROS\n" );
        Console.Write      ( "\nFORNECER O NUMERO DE VEZES (N): " );
        N = int.Parse      ( Console.ReadLine ( ) );
        for( CONTADOR = 1; CONTADOR <= N; CONTADOR++ )
        {
            Console.Write    ( "\n" + CONTADOR + " FORNECER UM VALOR INTEIRO: " );
            X = int.Parse    ( Console.ReadLine ( ) );
            Console.WriteLine ( "\nO VALOR DIGITADO FOI " + X );
        } // PARA CONTADOR EM [1:N]
        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO406
//
// metodos auxiliares
using System;

class CS0406
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E IMPRIMIR 03 VALORES INTEIROS
        // VARIAVEIS:
        int X,
            CONTADOR;

        Console.WriteLine ( "EXEMPLO406 - LER E IMPRIMIR 03 VALORES INTEIROS\n" );
        CONTADOR = 1;
        do
            // REPETIR
        {
            Console.Write    ( "\n" + CONTADOR + " DIGITE UM VALOR INTEIRO: " );
            X = int.Parse    ( Console.ReadLine ( ) );
            Console.WriteLine ( "\nO VALOR DIGITADO FOI " + X );
            CONTADOR = CONTADOR + 1;
        }
        while( CONTADOR <= 3 ); // ATE' ( CONTADOR > 3 )
        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```



```

/*
//
// ----- EXEMPLO407
//
// metodos auxiliares
using System;

class CS0407
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
        // VARIÁVEIS:
        int X, CONTADOR;

        Console.WriteLine ( "EXEMPLO407 - LER E IMPRIMIR (N) VALORES INTEIROS\n" );
        Console.Write ( "\nFORNECER O NUMERO DE VEZES (N): " );
        CONTADOR = int.Parse( Console.ReadLine ( ) );
        do
            // REPETIR
        {
            Console.Write ( "\n" + CONTADOR + " FORNECER UM VALOR INTEIRO : " );
            X = int.Parse ( Console.ReadLine ( ) );
            Console.WriteLine ( "\nO VALOR DIGITADO FOI " + X );
            CONTADOR = CONTADOR - 1;
        }
        while( CONTADOR > 0 ); // ATE' ( CONTADOR <= 3 )
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO408
//
// metodos auxiliares
using System;

class CS0408
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E IMPRIMIR INTEIROS DIFERENTES DE ZERO
        // VARIÁVEL:
        int X;

        Console.WriteLine ( "EXEMPLO408 - LER E IMPRIMIR INTEIROS NAO NULOS\n" );
        Console.Write ( "\nFORNECER UM VALOR INTEIRO (0 = PARAR): " );
        X = int.Parse ( Console.ReadLine ( ) );
        while( X != 0 ) // REPETIR
        {
            Console.WriteLine ( "\nO VALOR DIGITADO FOI " + X );
            Console.Write ( "\nDIGITE UM VALOR INTEIRO QUALQUER: " );
            X = int.Parse ( Console.ReadLine ( ) );
        } // ENQUANTO X DIFERENTE DE ZERO
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
// metodos auxiliares
// ----- EXEMPLO409
using System;

class CS0409
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER UM INTEIRO DIFERENTE DE ZERO
        // VARIABEL:
        int X;

        Console.WriteLine ( "EXEMPLO409 - PARA LER UM INTEIRO NAO NULO\n" );
        Console.Write ( "\nFORNECER UM VALOR DIFERENTE DE ZERO: " );
        X = int.Parse ( Console.ReadLine ( ) );
        while( X == 0 )    // REPETIR
        {
            Console.Write ( "\nFORNECER UM VALOR DIFERENTE DE ZERO: " );
            X = int.Parse ( Console.ReadLine ( ) );
        }    // ENQUANTO X IGUAL A ZERO
        Console.WriteLine ( "\nDIGITADO UM NUMERO DIFERENTE DE ZERO\n" );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
// ----- EXEMPLO410
// metodos auxiliares
using System;

class CS0410
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER UM INTEIRO DIFERENTE DE ZERO
        // VARIABEL :
        int X;

        Console.WriteLine ( "EXEMPLO410 - LER UM INTEIRO NAO NULO\n" );
        do
            // REPETIR
        {
            Console.Write ( "\nFORNECER UM VALOR DIFERENTE DE ZERO: " );
            X = int.Parse ( Console.ReadLine ( ) );
        }
        while( X == 0 );    // ATE' X DIFERENTE DE ZERO
        Console.WriteLine ( "\nDIGITADO UM NUMERO DIFERENTE DE ZERO\n" );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

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

/**/
//
// ----- EXEMPLO501
//
// metodos auxiliares
using System;

class CS0501
{
    public static void P1 ( )
    {
        Console.WriteLine ( "\nCHAMADO O PROCEDIMENTO P1 SEM PARAMETROS" );
        Console.WriteLine ( );
    } // fim procedimento P1 ( )

    public static void Main ( )
    {
        // PROGRAMA PARA LIDAR COM PROCEDIMENTO
        Console.WriteLine ( "EXEMPLO0501 - CHAMADA A UM PROCEDIMENTO" );
        P1( );           // chamada ao procedimento

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
/**/
/*
//
// ----- EXEMPLO502
//
// metodos auxiliares
using System;

class CS0502
{
    static int X;          // VARIABEL GLOBAL PARA A CLASSE

    public static void P1 ( )
    {
        Console.WriteLine ( "\nCHAMADO O PROCEDIMENTO P1 SEM PARAMETROS" );
        Console.WriteLine ( );
    } // fim procedimento P1 ( )

    public static void Main ( )
    {
        // PROGRAMA PARA LIDAR COM PROCEDIMENTO
        Console.WriteLine ( "EXEMPLO0502 - CHAMADA COM VARIABEL GLOBAL\n" );
        for ( X = 1; X <= 5; X = X + 1 )
            P1( );          // chamar 5 vezes

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO503
//
// metodos auxiliares
using System;

class CS0503
{
    static int X;          // VARIAVEL GLOBAL PARA A CLASSE

    public static void P1 ( )
    {
        X = X + 1;
        Console.WriteLine ( "\nCHAMADO O PROCEDIMENTO P1 " + X + " VEZ(ES)" );
        if( X < 5 ) P1 ( );    // chamar recursivamente
        Console.Write ( "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " + X );
        X = X - 1;
        Console.ReadLine ( );
    } // fim procedimento P1 ( )

    public static void Main ( )
    {
        // PROGRAMA PARA LIDAR COM PROCEDIMENTO RECURSIVO
        Console.WriteLine ( "EXEMPLO0503 - RECURSAO COM CONTROLE GLOBAL\n" );
        X = 0;
        P1 ( );          // OBSERVAR A RECURSIVIDADE !
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO504
//
// metodos auxiliares
using System;

class CS0504
{
    public static void P1 ( int X )
    {
        Console.WriteLine ( "\nCHAMADO O PROCEDIMENTO P1 " + X + " VEZ(ES)" );
        if( X < 5 )
            P1 ( X + 1 );    // chamar recursivamente com parametro
        Console.Write ( "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " + X );
        Console.ReadLine ( );
    } // fim procedimento P1 ( )

    public static void Main ( )
    {
        // PROGRAMA PARA LIDAR COM PROCEDIMENTO RECURSIVO COM PARAMETRO
        Console.WriteLine ( "EXEMPLO0504 - RECURSAO COM PARAMETRO\n" );
        P1 ( 1 );          // OBSERVAR A RECURSIVIDADE !
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO505
//
// metodos auxiliares
using System;

class CS0505
{
    public static void P1 ( int X )
    {
        Console.WriteLine ( "\nCHAMADO O PROCEDIMENTO P1 " + X + " VEZ(ES)" );
        if( X > 1 )
            P1( X - 1 );    // chamar recursivamente com parametro
        Console.Write ( "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " + X );
        Console.ReadLine ( );
    } // fim procedimento P1 ( )

    public static void Main ( )
    {
        // PROGRAMA PARA LIDAR COM PROCEDIMENTO RECURSIVO COM PARAMETRO
        Console.WriteLine ( "EXEMPLO0505 - RECURSAO COM PARAMETRO\n" );
        P1( 5 );           // OBSERVAR A RECURSIVIDADE !

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO506
//
// metodos auxiliares
using System;

class CS0506
{
    public static void P1 ( int X )
    {
        Console.WriteLine ( "\nCHAMADO O PROCEDIMENTO P1 " + X + " VEZ(ES)" );
        if( X < 5 )
            P2( X );          // chamar recursivamente com parametro
        Console.Write ( "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " + X );
        Console.ReadLine ( );
    } // fim procedimento P1 ( )

    public static void P2 ( int X )
    {
        Console.WriteLine ( "\nCHAMADO O PROCEDIMENTO P2 COM X = " + X );
        X = X + 1;
        Console.Write ( "\nCHAMAR O PROCEDIMENTO P1 COM X = " + X );
        Console.ReadLine ( );
        P1 ( X );
    } // fim procedimento P2 ( )

    public static void Main ( )
    {
        // PROGRAMA PARA LIDAR COM PROCEDIMENTO RECURSIVO INDIRETO
        Console.WriteLine ( "EXEMPLO0506 - CHAMADA RECURSIVA INDIRETA\n" );
        P1( 1 );          // OBSERVAR A RECURSIVIDADE INDIRETA!

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO507
//
// metodos auxiliares
using System;

class CS0507
{
    public static void P1 ( ref int X )
    {
        X = X + 1;
        Console.WriteLine ( "\nCHAMADO O PROCEDIMENTO P1 " + X + " VEZ(ES)" );
        if( X < 5 ) P1( ref X );          // chamar recursivamente
        Console.Write ( "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " + X );
        X = X - 1;
        Console.ReadLine ( );
    } // fim procedimento P1 ( )

    public static void Main ( )
    {
        // VARIÁVEL LOCAL:
        int X;

        // PROGRAMA PARA LIDAR COM PROCEDIMENTO RECURSIVO
        Console.WriteLine ( "EXEMPLO0507 - RECURSAO " +
                           "COM PASSAGEM POR REFERENCIA\n" );

        X = 0;
        P1( ref X );      // OBSERVAR A REPETICAO INFINITA !

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO508
//
// metodos auxiliares
using System;

class CS0508
{
    public static void P1 ( int X )
    {
        Console.WriteLine ( "\nCHAMADO O PROCEDIMENTO P1 " + X + " VEZ(ES)" );
        if( X < 4 )
        {
            P1( X + 1 );
            P2( X + 1 );
        } // fim se
        Console.Write ( "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " + X );
        Console.ReadLine ( );
    } // fim procedimento P1 ( )

    public static void P2 ( int X )
    {
        Console.WriteLine ( "\nCHAMADO O PROCEDIMENTO P2 COM X = " + X );
        if( X > 1 )
            P2 ( X - 1 );
        Console.Write ( "\nRETORNANDO AO PROCEDIMENTO P2 PARA A CHAMADA " + X );
        Console.ReadLine ( );
    } // fim procedimento P2 ( )

    public static void Main ( )
    {
        // PROGRAMA PARA LIDAR COM PROCEDIMENTO RECURSIVO MULTIPLO
        Console.WriteLine ( "EXEMPLO0508 - MULTIPLAS CHAMADAS/RETORNOS\n" );
        P1( 1 );           // OBSERVAR A RECURSIVIDADE INDIRETA!

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```



```

/*
//
// ----- EXEMPLO509
//

class UTIL01
{
    public static void LIMPAR_TELA ( )
    {
        System.Console.Clear ( );
    } // fim da acao LIMPAR_TELA ( )

    public static void ESPERAR ( )
    {
        System.Console.ReadLine ( );
    } // fim da acao ESPERAR ( )

} // fim da classe

class UTIL02
{
    public static void ESCREVER_CARACTERES ( string S )
    {
        System.Console.Write ( S );
    } // fim da acao ESCREVER_CARACTERES ( )

} // fim da classe

class CS0509
{
    public static void Main ( )
    {
        // PROGRAMA PARA MOSTRAR O USO DE MULTIPLAS CLASSES
        UTIL01.LIMPAR_TELA ( );
        UTIL02.ESCREVER_CARACTERES ( "EXEMPLO0509 - MULTIPLAS CLASSES\n" );
        System.Console.Write ( "\nApertar ENTER para terminar." );
        UTIL01.ESPERAR ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO510
//

namespace CS0510
{
    class UTIL01
    {
        public static void LIMPAR_TELA ( )
        {
            System.Console.Clear ( );
        } // fim da acao LIMPAR_TELA ( )

        public static void ESPERAR ( )
        {
            System.Console.ReadLine ( );
        } // fim da acao ESPERAR ( )

    } // fim da classe

    class UTIL02
    {
        public static void ESCREVER_CARACTERES ( string S )
        {
            System.Console.Write ( S );
        } // fim da acao ESCREVER_CARACTERES ( )

    } // fim da classe

    class CS0510
    {
        public static void Main ( )
        {
            // PROGRAMA PARA MOSTRAR O USO DE CLASSES AGRUPADAS
            UTIL01.LIMPAR_TELA ( );
            UTIL02.ESCREVER_CARACTERES( "EXEMPLO0510 - CLASSES AGRUPADAS\n" );
            System.Console.Write ( "\nApertar ENTER para terminar." );
            UTIL01.ESPERAR ( );
        } // fim da acao principal
    } // fim da classe
} // fim do grupo de classes
*/

```

```

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

/**/
//
// ----- EXEMPLO601
//
// metodos auxiliares
using System;

class CS0601
{
    public static void CONTAR ( int X )
    {
        if( X > 0 )
        {
            CONTAR ( X-1 );
            Console.WriteLine ( "\n " + X );
        }
    } // fim procedimento CONTAR( )

    public static void Main ( )
    {
        // PROGRAMA PARA CONTAR DE 1 ATE' 5, RECURSIVAMENTE
        Console.WriteLine ( "EXEMPLO601 - CONTAR DE 1 A 5 RECURSIVAMENTE\n" );
        CONTAR ( 5 );

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
/**/

```

```

/*
//
// ----- EXEMPLO602
//
// metodos auxiliares
using System;

class CS0602
{
    public static void CONTAR ( int X )
    {
        if( X > 0 )
        {
            Console.WriteLine ( "\n " + X );  CONTAR ( X-1 );
        }
    } // fim procedimento CONTAR( )

    public static void Main ( )
    {
        // PROGRAMA PARA CONTAR DE 5 ATE' 1, RECURSIVAMENTE
        Console.WriteLine ( "EXEMPLO602 - CONTAR DE 5 A 1 RECURSIVAMENTE\n" );
        CONTAR ( 5 );

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
/*
//
// ----- EXEMPLO603
//
// metodos auxiliares
using System;

class CS0603
{
    public static void PARES ( int X )
    {
        if ( X > 0 )
        {
            if ( X % 2 == 0 )
            {
                PARES ( X-2 );  Console.WriteLine ( "\n " + X );
            }
            else
                PARES ( X-1 );
        }
    } // fim procedimento PARES ( )

    public static void Main ( )
    {
        // PROGRAMA PARA MOSTRAR PARES, RECURSIVAMENTE
        Console.WriteLine ( "EXEMPLO603 - MOSTRAR OS PARES <= 10\n" );
        PARES ( 10 );

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```
/*
//
// ----- EXEMPLO604
//
// metodos auxiliares
using System;

class CS0604
{
    public static void PARES ( int X )
    {
        if ( X > 0 )
        {
            PARES ( X-1 );
            Console.WriteLine ( "\n " + (2*X) );
        }
    } // fim procedimento PARES ( )

    public static void Main ( )
    {
        // PROGRAMA PARA MOSTRAR PARES, RECURSIVAMENTE
        Console.WriteLine ( "EXEMPLO604 - MOSTRAR OS PARES <= 10\n" );
        PARES ( 5 );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/
```

```

/*
//
// ----- EXEMPLO605
//
// metodos auxiliares
using System;

class CS0605
{
    public static void PARES ( int X, ref int S )
    {
        if ( X > 0 )
        {
            PARES ( X-1, ref S );
            S = S + 2*X;
        }
        else
        {
            S = 0;
        }
    } // fim procedimento PARES ( )

    public static void Main ( )
    {
        // PROGRAMA PARA SOMAR OS 5 PRIMEIROS PARES, RECURSIVAMENTE
        // DADO:
        int SOMA = 0;

        Console.WriteLine ( "EXEMPLO605 - SOMAR OS 5 PRIMEIROS PARES\n" );
        PARES ( 5, ref SOMA );
        Console.WriteLine ( "\nSOMA DOS 5 PRIMEIROS PARES = " + SOMA );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO606
//
// metodos auxiliares
using System;

class CS0606
{
    public static int PARES ( int X )
    {
        // DADO LOCAL:
        int S;

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

        return ( S );
    } // fim funcao PARES ( )

    public static void Main ( )
    {
        // PROGRAMA PARA SOMAR OS 5 PRIMEIROS PARES, RECURSIVAMENTE
        // DADO:
        int SOMA;

        Console.WriteLine ( "EXEMPLO606 - SOMAR OS 5 PRIMEIROS PARES\n" );
        SOMA = PARES ( 5 );
        Console.WriteLine ( "\nSOMA DOS 5 PRIMEIROS PARES = " + SOMA );

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO607
//
// metodos auxiliares
using System;

class CS0607
{
    public static int PARES ( int X )
    {
        // DADO LOCAL:
        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 ( )

    public static void Main ( )
    {
        // PROGRAMA PARA CONTAR PARES, RECURSIVAMENTE

        Console.WriteLine ( "EXEMPLO607 - CONTAR PARES\n" );
        Console.WriteLine ( "\nPARES <= 10 = " + PARES ( 10 ) );

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```



```

/*
//
// ----- EXEMPLO608
//
// metodos auxiliares
using System;

class CS0608
{
    public static bool PROCURAR ( char LETRA, string S, int POSICAO )
    {
        // DADO LOCAL:
        bool R;

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

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

    public static void Main ( )
    {
        // PROGRAMA RECURSIVO PARA ACHAR A POSICAO DE UMA LETRA
        // DADOS:
        string S;
        char L;

        Console.WriteLine ( "EXEMPLO608 - PROCURAR LETRA EM UMA SENTENCA\n" );
        Console.Write ( "\nFORNECER UMA SENTENCA: " );
        S = Console.ReadLine ( );
        Console.Write ( "\nFORNECER UMA LETRA PARA SER PROCURADA: " );
        L = (char) Console.Read ( );
        Console.ReadLine ( ); // limpar o buffer
        Console.WriteLine ( "\nRESPOSTA = " + PROCURAR(L,S,0) );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO609
//
// metodos auxiliares
using System;

class CS0609
{
    public static int PROCURAR ( char LETRA, string S, int POSICAO )
    {
        // DADO LOCAL:
        int R;

        if ( POSICAO < S.Length )
            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 ( )

    public static void Main ( )
    {
        // PROGRAMA RECURSIVO PARA ACHAR A POSICAO DE UMA LETRA
        // DADOS:
        string S;
        char L;

        Console.WriteLine ( "EXEMPLO609 - POSICAO DE UMA LETRA\n" );
        Console.Write      ( "\nFORNECER UMA SENTENCA: " );
        S = Console.ReadLine ( );
        Console.Write      ( "\nFORNECER UMA LETRA PARA SER PROCURADA: " );
        L = (char) Console.Read ( );
        Console.ReadLine ( ); // limpar o buffer
        Console.WriteLine ( "\nRESPOSTA = " + PROCURAR(L,S,0) );

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO610
//
// metodos auxiliares
using System;

class CS0610
{
    public static int PROCURAR ( char LETRA, string S, int POSICAO )
    {
        // DADO LOCAL:
        int R;

        if ( POSICAO < S.Length )
            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 ( )

    public static void Main ( )
    {
        // PROGRAMA RECURSIVO PARA PROCURAR OCORRENCIAS DE UMA LETRA
        // DADOS:
        string S;
        char L;

        Console.WriteLine ( "EXEMPLO610 - PROCURAR OCORRENCIAS DE LETRA\n" );
        Console.Write      ( "\nFORNECER UMA SENTENCA: " );
        S = Console.ReadLine ( );
        Console.Write      ( "\nFORNECER UMA LETRA PARA SER PROCURADA: " );
        L = (char) Console.Read ( );
        Console.ReadLine ( ); // limpar o buffer
        Console.WriteLine ( "\nRESPOSTA = " + PROCURAR(L,S,0) );

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

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

/**/
//
// ----- EXEMPLO701
//
// metodos auxiliares
using System;

class CS0701
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER UMA TABELA DE INTEIROS
        // VARIAVEIS:
        int [ ] V = new int [ 10 ];
        int    X;

        Console.WriteLine ( "EXEMPLO701 - LER UMA TABELA DE 10 INTEIROS\n" );
            // REPETIR PARA CADA POSICAO
        for ( X = 0; X < 10; X++ )
        {
            // a primeira posicao e' zero !
            Console.Write ( "FORNECER O " + (X+1) + "o. INTEIRO: " );
            V [ X ] = int.Parse ( Console.ReadLine ( ) );
        }
            // FIM REPETIR
        Console.WriteLine ( "\nVETOR LIDO: \n" );
            // REPETIR PARA CADA POSICAO
        for ( X = 0; X < 10; X++ )
        {
            Console.Write ( V[ X ] + " " );
        }
            // FIM REPETIR

        Console.WriteLine ( );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
/**/

```

```

/*
//
// ----- EXEMPLO702
//
// metodos auxiliares
using System;

class CS0702
{
    public static void Main ( )
    {
        // PROGRAMA PARA SOMAR UMA TABELA DE INTEIROS
        // VARIAVEIS:
        int [ ] V = new int [ 10 ];
        int    X, SOMA;

        Console.WriteLine ( "EXEMPLO702 - SOMAR UMA TABELA COM 10 INTEIROS\n" );
        // REPETIR PARA CADA POSICAO
        for ( X = 0; X < 10; X++ )
        {
            // a primeira posicao e' zero !
            Console.Write ( "FORNECER O " + (X+1) + "o. INTEIRO: " );
            V [ X ] = int.Parse ( Console.ReadLine ( ) );
        }
        // FIM REPETIR
        Console.WriteLine ( "\nVETOR LIDO: \n" );
        // REPETIR PARA CADA POSICAO
        SOMA = 0;
        for ( X = 0; X < 10; X++ )
        {
            SOMA = SOMA + V[ X ];
        }
        // FIM REPETIR

        Console.WriteLine ( "SOMA = " + SOMA );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO703
//
// metodos auxiliares
using System;

class CS0703
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A MEDIA DE UMA TABELA DE INTEIROS
        // VARIAVEIS:
        int [ ] V = new int [ 10 ];
        int    X, SOMA;
        double MEDIA;

        Console.WriteLine ( "EXEMPLO703 - MEDIA UMA TABELA COM 10 INTEIROS\n" );
        // REPETIR PARA CADA POSICAO
        for ( X = 0; X < 10; X++ )
        {
            // a primeira posicao e' zero !
            Console.Write ( "FORNECER O " + (X+1) + "o. INTEIRO: " );
            V [ X ] = int.Parse ( Console.ReadLine ( ) );
        }
        // FIM REPETIR
        Console.WriteLine ( "\nVETOR LIDO: \n" );
        // REPETIR PARA CADA POSICAO

        SOMA = 0;
        for ( X = 0; X < 10; X++ )
        {
            SOMA = SOMA + V[ X ];
        }
        // FIM REPETIR
        MEDIA = SOMA / 10.0;
        Console.WriteLine ( "MEDIA = " + MEDIA );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO704
//
// metodos auxiliares
using System;

class CS0704
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER UMA PALAVRA
        // VARIAVEIS:
        string PALAVRA;
        int X;

        Console.WriteLine ( "EXEMPLO704 - DECOMPOR UMA PALAVRA\n" );
        Console.Write ( "\nFORNECER UMA PALAVRA: " );
        PALAVRA = Console.ReadLine ( );

        Console.Write ( "\nLETRAS DA PALAVRA LIDA: " );
        // REPETIR PARA CADA POSICAO
        for( X = 0; X < PALAVRA.Length; X++ )
            Console.Write ( PALAVRA[ X ] + " " );

        Console.WriteLine ( );
        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO705
//
// metodos auxiliares
using System;

class CS0705
{
    public static void Main ( )
    {
        // PROGRAMA PARA PROCURAR LETRA EM UMA PALAVRA
        // DADOS:
        string PALAVRA;
        char  LETRA;
        int   X;
        bool  ACHAR;

        Console.WriteLine ( "EXEMPLO705 - PROCURAR LETRA EM UMA PALAVRA\n" );
        Console.Write      ( "\nFORNECER UMA PALAVRA: " );
        PALAVRA = Console.ReadLine ( );
        Console.Write      ( "\nFORNECER UMA LETRA PARA SER PROCURADA: " );
        LETRA = (char) Console.Read ( );
        Console.ReadLine ( );// limpar o buffer

        ACHAR = false;
        X     = 0;

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

        if ( ACHAR )
            Console.WriteLine ( "\nLETRA ENCONTRADA" );
        else
            Console.WriteLine ( "\nLETRA NAO ENCONTRADA" );

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```



```

/*
//
// ----- EXEMPLO706
//
// metodos auxiliares
using System;

class CS0706
{
    public static void Main ( )
    {
        // PROGRAMA PARA AVALIAR UM POLINOMIO
        // VARIAVEIS:
        int [ ] P = new int [ 10 ];
        int    Y, N;
        double X, PX;

        Console.WriteLine ( "EXEMPLO706 - AVALIAR UM POLINOMIO\n" );

        Console.Write ( "\nFORNECER O GRAU DO POLINOMIO: " );
        N = int.Parse ( Console.ReadLine ( ) );
        // REPETIR PARA CADA POSICAO
        for( Y = 0; Y <= N; Y++ )
        {
            Console.Write ( "\nFORNECER O " + (Y+1) + "o COEFICIENTE: " );
            P [ Y ] = int.Parse ( Console.ReadLine ( ) );
        }
        // FIM REPETIR

        Console.Write ( "\nFORNECER O PONTO DE AVALIACAO: " );
        X = int.Parse ( Console.ReadLine ( ) );

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

        Console.WriteLine ( "\nP" + X + " = " + PX );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO707
//
// metodos auxiliares
using System;

class CS0707
{
    public static void Main ( )
    {
        // PROGRAMA PARA AVALIAR O COMPRIMENTO DE UM VETOR
        // VARIAVEIS:
        int [ ] V = new int [ 3 ];
        int    X;
        double SOMA;

        Console.WriteLine ( "EXEMPLO707 - COMPRIMENTO DE UM VETOR\n" );

        Console.Write ( "\nFORNECER O VALOR DE X : " );
        V [ 0 ] = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y : " );
        V [ 1 ] = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z : " );
        V [ 2 ] = int.Parse ( Console.ReadLine ( ) );

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

        Console.WriteLine ( "\nCOMPRIMENTO = " + Math.Sqrt ( SOMA ) );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO708
//
// metodos auxiliares
using System;

class CS0708
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER E MOSTRAR UMA MATRIZ
        // VARIAVEIS:
        int [,] M = new int [2, 2];
        int    X, Y;

        Console.WriteLine ( "EXEMPLO708 - LER E MOSTRAR UMA MATRIZ 2x2\n" );

        // REPETIR PARA CADA LINHA
        for ( X = 0; X < 2; X++ )
        {
            // REPETIR PARA CADA COLUNA
            for( Y = 0; Y < 2; Y++ )
            {
                Console.Write ( "\nFORNECER ELEMENTO " + (X+1) + "," + (Y+1) + " : " );
                M [ X, Y ] = int.Parse ( Console.ReadLine ( ) );
            }
            // FIM REPETIR
        }
        // FIM REPETIR

        Console.WriteLine ( "\nMatriz lida:\n" );

        // REPETIR PARA CADA LINHA
        for ( X = 0; X < 2; X++ )
        {
            // REPETIR PARA CADA COLUNA
            for( Y = 0; Y < 2; Y++ )
            {
                Console.Write ( M [ X, Y ] + "\t" );
            }
            // FIM REPETIR
            Console.WriteLine ( );
        }
        // FIM REPETIR

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO709
//
// metodos auxiliares
using System;

class CS0709
{
    public static void Main ( )
    {
        // PROGRAMA PARA MONTAR A TRANSPOSTA DE UMA MATRIZ
        // VARIAVEIS:
        int [,] M = new int [2, 2], // MATRIZ ORIGINAL
            MT = new int [2, 2]; // MATRIZ TRANSPOSTA
        int X, Y;

        Console.WriteLine ( "EXEMPLO709 - LER UMA MATRIZ 2x2\n" );

        // REPETIR PARA CADA LINHA
        for ( X = 0; X < 2; X++ )
        {
            // REPETIR PARA CADA COLUMA
            for( Y = 0; Y < 2; Y++ )
            {
                Console.Write ( "\nFORNECER ELEMENTO " + (X+1) + "," + (Y+1) + " : " );
                M [ X, Y ] = int.Parse ( Console.ReadLine ( ) );
                MT[ Y, X ] = M [ X, Y ];
            } // FIM REPETIR
        } // FIM REPETIR

        Console.WriteLine ( "\nMatriz transposta:\n" );

        // REPETIR PARA CADA LINHA
        for ( X = 0; X < 2; X++ )
        {
            // REPETIR PARA CADA COLUMA
            for( Y = 0; Y < 2; Y++ )
            {
                Console.Write ( MT [ X, Y ] + "\t" );
            } // FIM REPETIR
            Console.WriteLine ( );
        } // FIM REPETIR

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO710
//
// metodos auxiliares
using System;

class CS0710
{
    const int ORDEM = 3;

    public static void Main ( )
    {
        // PROGRAMA PARA MOSTRAR A DIAGONAL DE UMA MATRIZ
        // VARIAVEIS:
        int [,] M = new int [ ORDEM, ORDEM ];
        int    X, Y;

        Console.WriteLine ( "EXEMPLO7108 - MOSTRAR A DIAGONAL DE UMA MATRIZ\n" );

        // REPETIR PARA CADA LINHA
        for ( X = 0; X < ORDEM; X++ )
        {
            // REPETIR PARA CADA COLUMA
            for( Y = 0; Y < ORDEM; Y++ )
            {
                Console.Write ( "\nFORNECER ELEMENTO " + (X+1) + "," + (Y+1) + " : " );
                M [ X, Y ] = int.Parse ( Console.ReadLine ( ) );
            }
            // FIM REPETIR
        }
        // FIM REPETIR

        Console.WriteLine ( "\nDiagonal da matriz lida:\n" );

        // REPETIR PARA CADA LINHA
        for ( X = 0; X < ORDEM; X++ )
        {
            // REPETIR PARA CADA COLUMA
            for ( Y = 0; Y < ORDEM; Y++ )
            {
                if ( X == Y ) // SE ESTIVER NA DIAGONAL
                    Console.Write ( M [ X, Y ] + "\t" );
            }
            // FIM REPETIR
        }
        // FIM REPETIR

        Console.WriteLine ( );
        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

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

/**/
//
// ----- EXEMPLO801
//
// metodos auxiliares
using System;

class PONTOS
{
    public int X, Y, Z;
} // fim da classe

class CS0801
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
        // DADOS:
        PONTOS P1 = new PONTOS ( ),
                P2 = new PONTOS ( ),
                P3 = new PONTOS ( );
        double D;

        Console.WriteLine ( "EXEMPLO801 - DISTANCIA ENTRE PONTOS\n" );
        Console.WriteLine ( "\nENTRE COM O PRIMEIRO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        P1.X = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        P1.Y = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        P1.Z = int.Parse ( Console.ReadLine ( ) );

        Console.WriteLine ( "\nENTRE COM O SEGUNDO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        P2.X = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        P2.Y = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        P2.Z = int.Parse ( Console.ReadLine ( ) );

        P3.X = P2.X - P1.X;
        P3.Y = P2.Y - P1.Y;
        P3.Z = P2.Z - P1.Z;
        D = Math.Sqrt ( Math.Pow ( P3.X, 2.0 )+
                        Math.Pow ( P3.Y, 2.0 )+
                        Math.Pow ( P3.Z, 2.0 ) );
        Console.WriteLine ( "\nDISTANCIA = " + D );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
/**/

```

```

/*
//
// ----- EXEMPLO802
//
// metodos auxiliares
using System;

class PONTOS
{
    public int X, Y, Z;
} // fim da classe

class CS0802
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
        // DADOS:
        PONTOS P1 = new PONTOS ( ),
        P2 = new PONTOS ( );
        double D;

        Console.WriteLine ( "EXEMPLO802 - DISTANCIA ENTRE PONTOS\n" );
        Console.WriteLine ( "\nENTRE COM O PRIMEIRO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        P1.X = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        P1.Y = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        P1.Z = int.Parse ( Console.ReadLine ( ) );

        Console.WriteLine ( "\nENTRE COM O SEGUNDO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        P2.X = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        P2.Y = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        P2.Z = int.Parse ( Console.ReadLine ( ) );

        D = Math.Sqrt ( Math.Pow ( P2.X-P1.X, 2.0 )+
                        Math.Pow ( P2.Y-P1.Y, 2.0 )+
                        Math.Pow ( P2.Z-P1.Z, 2.0 ) );
        Console.WriteLine ( "\nDISTANCIA = " + D );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO803
//
// metodos auxiliares
using System;

class PONTOS
{
    public int X, Y, Z;
} // fim da classe

class CS0803
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
        // DADOS:
        PONTOS [ ] V = new PONTOS [ 2 ];
        V [ 0 ] = new PONTOS ( );
        V [ 1 ] = new PONTOS ( );
        double D;

        Console.WriteLine ( "EXEMPLO803 - DISTANCIA ENTRE PONTOS\n" );
        Console.WriteLine ( "\nENTRE COM O PRIMEIRO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        V [ 0 ].X = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        V [ 0 ].Y = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        V [ 0 ].Z = int.Parse ( Console.ReadLine ( ) );

        Console.WriteLine ( "\nENTRE COM O SEGUNDO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        V [ 1 ].X = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        V [ 1 ].Y = int.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        V [ 1 ].Z = int.Parse ( Console.ReadLine ( ) );

        D = Math.Sqrt ( Math.Pow ( V[1].X-V[0].X, 2.0 )+
                        Math.Pow ( V[1].Y-V[0].Y, 2.0 )+
                        Math.Pow ( V[1].Z-V[0].Z, 2.0 ) );
        Console.WriteLine ( "\nDISTANCIA = " + D );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```



```

/*
//
// ----- EXEMPLO804
//
// metodos auxiliares
using System;

class PONTOS
{
    public double [ ] P1 = new double [ 3 ];
    public double [ ] P2 = new double [ 3 ];
} // fim da classe

class CS0804
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
        // DADOS:
        PONTOS V = new PONTOS ( );
        double D;

        Console.WriteLine ( "EXEMPLO804 - DISTANCIA ENTRE PONTOS\n" );
        Console.WriteLine ( "\nENTRE COM O PRIMEIRO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        V.P1 [ 0 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        V.P1 [ 1 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        V.P1 [ 2 ] = double.Parse ( Console.ReadLine ( ) );

        Console.WriteLine ( "\nENTRE COM O SEGUNDO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        V.P2 [ 0 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        V.P2 [ 1 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        V.P2 [ 2 ] = double.Parse ( Console.ReadLine ( ) );

        D = Math.Sqrt ( Math.Pow ( V.P1[0]-V.P2[0], 2.0 )+
                        Math.Pow ( V.P1[1]-V.P2[1], 2.0 )+
                        Math.Pow ( V.P1[2]-V.P2[2], 2.0 ) );
        Console.WriteLine ( "\nDISTANCIA = " + D );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO805
//
// metodos auxiliares
using System;

class PONTO
{
    public double X, Y, Z;
} // fim da classe

class PONTOS
{
    public PONTO P1 = new PONTO ( );
    public PONTO P2 = new PONTO ( );
} // fim da classe

class CS0805
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
        // DADOS:
        PONTOS P = new PONTOS ( );
        double D;

        Console.WriteLine ( "EXEMPLO805 - DISTANCIA ENTRE PONTOS\n" );
        Console.WriteLine ( "Entre com o primeiro ponto:" );
        Console.Write ( "\nFornecer o valor de X: " );
        P.P1.X = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFornecer o valor de Y: " );
        P.P1.Y = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFornecer o valor de Z: " );
        P.P1.Z = double.Parse ( Console.ReadLine ( ) );

        Console.WriteLine ( "Entre com o segundo ponto:" );
        Console.Write ( "\nFornecer o valor de X: " );
        P.P2.X = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFornecer o valor de Y: " );
        P.P2.Y = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFornecer o valor de Z: " );
        P.P2.Z = double.Parse ( Console.ReadLine ( ) );

        D = Math.Sqrt ( Math.Pow ( P.P2.X-P.P1.X, 2.0 )+
                        Math.Pow ( P.P2.Y-P.P1.Y, 2.0 )+
                        Math.Pow ( P.P2.Z-P.P1.Z, 2.0 ) );
        Console.WriteLine ( "\nDistancia = " + D );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO806
//
// metodos auxiliares
using System;

class PONTO
{
    public double X, Y, Z;
} // fim da classe

class PONTOS
{
    public PONTO P1 = new PONTO ( );
    public double [ ] P2 = new double [ 3 ];
} // fim da classe

class CS0806
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
        // DADOS:
        PONTOS P = new PONTOS ( );
        double D;

        Console.WriteLine ( "EXEMPLO806 - DISTANCIA ENTRE PONTOS\n" );
        Console.WriteLine ( "\nENTRE COM O PRIMEIRO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        P.P1.X = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        P.P1.Y = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        P.P1.Z = double.Parse ( Console.ReadLine ( ) );

        Console.WriteLine ( "\nENTRE COM O SEGUNDO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        P.P2 [ 0 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        P.P2 [ 1 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        P.P2 [ 2 ] = double.Parse ( Console.ReadLine ( ) );

        D = Math.Sqrt ( Math.Pow ( P.P2[0]-P.P1.X, 2.0 )+
                        Math.Pow ( P.P2[1]-P.P1.Y, 2.0 )+
                        Math.Pow ( P.P2[2]-P.P1.Z, 2.0 ) );
        Console.WriteLine ( "\nDISTANCIA = " + D );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO807
//
// metodos auxiliares
using System;

class PONTOS
{
    public double [,] P = new double [ 2, 3 ];
} // fim da classe

class CS0807
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
        // DADOS:
        PONTOS V = new PONTOS ( );
        double D;

        Console.WriteLine ( "EXEMPLO807 - DISTANCIA ENTRE PONTOS\n" );
        Console.WriteLine ( "\nENTRE COM O PRIMEIRO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        V.P [ 0, 0 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        V.P [ 0, 1 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        V.P [ 0, 2 ] = double.Parse ( Console.ReadLine ( ) );

        Console.WriteLine ( "\nENTRE COM O SEGUNDO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        V.P [ 1, 0 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        V.P [ 1, 1 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        V.P [ 1, 2 ] = double.Parse ( Console.ReadLine ( ) );

        D = Math.Sqrt ( Math.Pow ( V.P[1,0]-V.P[0,0], 2.0 )+
                        Math.Pow ( V.P[1,1]-V.P[0,1], 2.0 )+
                        Math.Pow ( V.P[1,2]-V.P[0,2], 2.0 ) );
        Console.WriteLine ( "\nDISTANCIA = " + D );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO808
//
// metodos auxiliares
using System;

class PONTO
{
    public double [] P1 = new double [ 3 ];
} // fim da classe

class PONTOS
{
    public PONTO [] P2 = new PONTO [ 2 ];
} // fim da classe

class CS0808
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
        // DADOS:
        PONTOS V = new PONTOS ( );
        V.P2 [ 0 ] = new PONTO ( );
        V.P2 [ 1 ] = new PONTO ( );

        double D;

        Console.WriteLine ( "EXEMPLO808 - DISTANCIA ENTRE PONTOS\n" );
        Console.WriteLine ( "\nENTRE COM O PRIMEIRO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        V.P2[0].P1[0] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        V.P2[0].P1[1] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        V.P2[0].P1[2] = double.Parse ( Console.ReadLine ( ) );

        Console.WriteLine ( "\nENTRE COM O SEGUNDO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        V.P2[1].P1[0] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        V.P2[1].P1[1] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        V.P2[1].P1[2] = double.Parse ( Console.ReadLine ( ) );

        D = Math.Sqrt ( Math.Pow ( V.P2[0].P1[0]-V.P2[1].P1[0], 2.0 )+
                        Math.Pow ( V.P2[0].P1[1]-V.P2[1].P1[1], 2.0 )+
                        Math.Pow ( V.P2[0].P1[2]-V.P2[1].P1[2], 2.0 ) );
        Console.WriteLine ( "\nDISTANCIA = " + D );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO809
//
// metodos auxiliares
using System;

class PONTO
{
    public double X, Y, Z;
} // fim da classe

class PONTOS
{
    public PONTO [ ] P2 = new PONTO [ 2 ];
} // fim da classe

class CS0809
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
        // DADOS:
        PONTOS V = new PONTOS ( );
        V.P2 [ 0 ] = new PONTO ( );
        V.P2 [ 1 ] = new PONTO ( );

        double D;

        Console.WriteLine ( "EXEMPLO809 - DISTANCIA ENTRE PONTOS\n" );
        Console.WriteLine ( "\nENTRE COM O PRIMEIRO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        V.P2[0].X = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        V.P2[0].Y = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        V.P2[0].Z = double.Parse ( Console.ReadLine ( ) );

        Console.WriteLine ( "\nENTRE COM O SEGUNDO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        V.P2[1].X = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        V.P2[1].Y = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        V.P2[1].Z = double.Parse ( Console.ReadLine ( ) );

        D = Math.Sqrt ( Math.Pow ( V.P2[0].X-V.P2[1].X, 2.0 )+
                        Math.Pow ( V.P2[0].Y-V.P2[1].Y, 2.0 )+
                        Math.Pow ( V.P2[0].Z-V.P2[1].Z, 2.0 ) );
        Console.WriteLine ( "\nDISTANCIA = " + D );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO810
//
// metodos auxiliares
using System;

class PONTO
{
    public double [ ] V = new double [ 3 ];
} // fim da classe

class PONTOS
{
    public PONTO P1 = new PONTO ( ),
        P2 = new PONTO ( );
} // fim da classe

class CS0810
{
    public static void Main ( )
    {
        // PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
        // DADOS:
        PONTOS P = new PONTOS ( );
        double D;

        Console.WriteLine ( "EXEMPLO810 - DISTANCIA ENTRE PONTOS\n" );
        Console.WriteLine ( "\nENTRE COM O PRIMEIRO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        P.P1.V [ 0 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        P.P1.V [ 1 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        P.P1.V [ 2 ] = double.Parse ( Console.ReadLine ( ) );

        Console.WriteLine ( "\nENTRE COM O SEGUNDO PONTO:" );
        Console.Write ( "\nFORNECER O VALOR DE X: " );
        P.P2.V [ 0 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Y: " );
        P.P2.V [ 1 ] = double.Parse ( Console.ReadLine ( ) );
        Console.Write ( "\nFORNECER O VALOR DE Z: " );
        P.P2.V [ 2 ] = double.Parse ( Console.ReadLine ( ) );

        D = Math.Sqrt ( Math.Pow ( P.P2.V [ 0 ]-P.P1.V [ 0 ], 2.0 )+
                        Math.Pow ( P.P2.V [ 1 ]-P.P1.V [ 1 ], 2.0 )+
                        Math.Pow ( P.P2.V [ 2 ]-P.P1.V [ 2 ], 2.0 ) );
        Console.WriteLine ( "\nDISTANCIA = " + D );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

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

/**/
//
// ----- EXEMPLO901
//
// metodos auxiliares
using System;
using System.IO;

class PONTO
{
    public int X, Y, Z;
} // fim da classe

class CS0901
{
    public static void Main ( )
    {
        // PROGRAMA PARA GRAVAR COORDENADAS DE PONTOS
        // DADOS:
        // ABRIR ARQUIVO PARA GRAVAR
        TextWriter A = new StreamWriter ( "PONTOS1.TXT" );

        PONTO P = new PONTO ( );
        int X;

        // REPETIR PARA DOIS PONTOS
        Console.WriteLine ( "EXEMPLO901 - GRAVAR COORDENADAS DE PONTOS\n" );
        for( X = 1; X <= 2; X++ )
        {
            Console.WriteLine ( "\nENTRE COM AS COORDENADAS DE UM PONTO:" );
            Console.Write ( "\nFORNECER O VALOR DE X: " );
            P.X = int.Parse ( Console.ReadLine ( ) );
            Console.Write ( "\nFORNECER O VALOR DE Y: " );
            P.Y = int.Parse ( Console.ReadLine ( ) );
            Console.Write ( "\nFORNECER O VALOR DE Z: " );
            P.Z = int.Parse ( Console.ReadLine ( ) );

            A.WriteLine ( P.X );
            A.WriteLine ( P.Y );
            A.WriteLine ( P.Z );
        } // FIM REPETIR
        A.Close ( ); // FECHAR ARQUIVO

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
/**/

```



```

/*
//
// ----- EXEMPLO902
//
// metodos auxiliares
using System;
using System.IO;

class PONTO
{
    public int X, Y, Z;
} // fim da classe

class CS0902
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER COORDENADAS DE PONTOS
        // DADOS:
        // ABRIR ARQUIVO PARA LER
        TextReader A = new StreamReader ( "PONTOS1.TXT" );

        PONTO P = new PONTO ( );
        int X;
        // REPETIR PARA DOIS PONTOS
        Console.WriteLine ( "EXEMPLO902 - LER COORDENADAS DE PONTOS\n" );
        for( X = 1; X <= 2; X++ )
        {
            P.X = int.Parse ( A.ReadLine ( ) );
            P.Y = int.Parse ( A.ReadLine ( ) );
            P.Z = int.Parse ( A.ReadLine ( ) );

            Console.WriteLine ( "\nPONTO " + X + " : " +
                                P.X + ", " + P.Y + ", " + P.Z );
        } // FIM REPETIR
        A.Close ( ); // FECHAR ARQUIVO

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO903
//
// metodos auxiliares
using System;
using System.IO;

class PONTO
{
    public int X, Y, Z;
} // fim da classe

class CS0903
{
    public static void Main ( )
    {
        // PROGRAMA PARA GRAVAR COORDENADAS DE PONTOS
        // DADOS:
            // ABRIR ARQUIVO PARA GRAVAR
        FileStream ARQUIVO = new FileStream ( "PONTOS2.TXT",
                                            FileMode.OpenOrCreate,
                                            FileAccess.Write );

        StreamWriter A    = new StreamWriter ( ARQUIVO );
        PONTO P = new PONTO ( );
        int  X;

            // REPETIR PARA DOIS PONTOS
        Console.WriteLine ( "EXEMPLO903 - GRAVAR COORDENADAS DE PONTOS\n" );
        for( X = 1; X <= 2; X++ )
        {
            Console.WriteLine ( "\nENTRE COM AS COORDENADAS DE UM PONTO:" );
            Console.Write   ( "\nFORNECER O VALOR DE X: " );
            P.X = int.Parse ( Console.ReadLine ( ) );
            Console.Write   ( "\nFORNECER O VALOR DE Y: " );
            P.Y = int.Parse ( Console.ReadLine ( ) );
            Console.Write   ( "\nFORNECER O VALOR DE Z: " );
            P.Z = int.Parse ( Console.ReadLine ( ) );

            A.Write ( P.X.ToString ("d3") + " " +
                    P.Y.ToString ("d3") + " " +
                    P.Z.ToString ("d3") );
            A.WriteLine ( );
        } // FIM REPETIR
        A.Close ( ); // FECHAR ARQUIVO

        Console.Write   ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO904
//
// metodos auxiliares
using System;
using System.IO;

class CS0904
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER COORDENADAS DE PONTOS
        // DADOS:
            // ABRIR ARQUIVO PARA LER
        FileStream ARQUIVO = new FileStream ( "PONTOS2.TXT",
                                            FileMode.OpenOrCreate,
                                            FileAccess.Read );
        StreamReader A    = new StreamReader ( ARQUIVO );

        string LINHA;
        int    X;

            // REPETIR PARA DOIS PONTOS
        Console.WriteLine ( "EXEMPLO904 - LER COORDENADAS DE PONTOS\n" );
        for( X = 1; X <= 2; X++ )
        {
            LINHA = A.ReadLine ( );
            Console.WriteLine ( "\nPONTO " + X + " : " + LINHA );
        }
        // FIM REPETIR
        A.Close ( ); // FECHAR ARQUIVO

        Console.Write      ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO905
//
// metodos auxiliares
using System;
using System.IO;

class PONTO
{
    public int X, Y, Z;
} // fim da classe

class CS0905
{
    public static void Main ( )
    {
        // PROGRAMA PARA COPIAR O ARQUIVO COM COORDENADAS DE PONTOS
        // DADOS:
            // ABRIR ARQUIVO PARA LER
        TextReader A1 = new StreamReader ( "PONTOS1.TXT" );
        TextWriter A2 = new StreamWriter ( "PONTOS3.TXT" );

        PONTO P = new PONTO ( );
        int X;

            // REPETIR PARA DOIS PONTOS
        Console.WriteLine ( "EXEMPLO905 - COPIAR ARQUIVO \n" );
        for( X = 1; X <= 2; X++ )
        {
            P.X = int.Parse ( A1.ReadLine ( ) );
            P.Y = int.Parse ( A1.ReadLine ( ) );
            P.Z = int.Parse ( A1.ReadLine ( ) );

            A2.WriteLine ( P.X );
            A2.WriteLine ( P.Y );
            A2.WriteLine ( P.Z );

            Console.WriteLine ( "\nCOPIADO PONTO " + X + " : " +
                                P.X + ", " + P.Y + ", " + P.Z );
        }
            // FIM REPETIR
        A1.Close ( ); // FECHAR ARQUIVO
        A2.Close ( ); // FECHAR ARQUIVO

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO906
//
// metodos auxiliares
using System;
using System.IO;

class PONTO
{
    public int X, Y, Z;
} // fim da classe

class CS0906
{
    public static void Main ( )
    {
        // PROGRAMA PARA ACRESCENTAR COORDENADAS DE PONTOS
        // DADOS:
            // ABRIR ARQUIVO PARA ACRESCENTAR
        FileStream ARQUIVO = new FileStream ( "PONTOS3.TXT",
                                            FileMode.Append );
        StreamWriter A    = new StreamWriter ( ARQUIVO );

        PONTO P = new PONTO ( );
        int  X;

            // REPETIR PARA DOIS PONTOS
        Console.WriteLine ( "EXEMPLO906 - ACRESCENTAR PONTOS\n" );
        for( X = 1; X <= 2; X++ )
        {
            Console.WriteLine ( "\nENTRE COM AS COORDENADAS DE UM PONTO:" );
            Console.Write   ( "\nFORNECER O VALOR DE X: " );
            P.X = int.Parse ( Console.ReadLine ( ) );
            Console.Write   ( "\nFORNECER O VALOR DE Y: " );
            P.Y = int.Parse ( Console.ReadLine ( ) );
            Console.Write   ( "\nFORNECER O VALOR DE Z: " );
            P.Z = int.Parse ( Console.ReadLine ( ) );

            A.WriteLine ( P.X );
            A.WriteLine ( P.Y );
            A.WriteLine ( P.Z );
        } // FIM REPETIR
        A.Close ( ); // FECHAR ARQUIVO

        Console.Write   ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO907
//
// metodos auxiliares
using System;
using System.IO;

class PONTO
{
    public int X, Y, Z;
} // fim da classe

class CS0907
{
    public static void Main ( )
    {
        // PROGRAMA PARA LER COORDENADAS DE PONTOS
        // DADOS:
            // ABRIR ARQUIVO PARA LER
        TextReader A = new StreamReader ( "PONTOS3.TXT" );

        PONTO P = new PONTO ( );
        string LINHA;
        int X;

            // REPETIR PARA DOIS PONTOS
        Console.WriteLine ( "EXEMPLO907 - LER COORDENADAS DE PONTOS\n" );
        X = 0;
        LINHA = A.ReadLine ( );
            // REPETIR ENQUANTO HOUVER DADOS
        while ( LINHA != null )
        {
            P.X = int.Parse ( LINHA );
            P.Y = int.Parse ( A.ReadLine ( ) );
            P.Z = int.Parse ( A.ReadLine ( ) );

            X = X + 1;
            Console.WriteLine ( "\nPONTO " + X + " : " +
                                P.X + ", " + P.Y + ", " + P.Z );
            LINHA = A.ReadLine ( );
        } // FIM REPETIR
        A.Close ( ); // FECHAR ARQUIVO

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO908
//
// metodos auxiliares
using System;
using System.IO;

class PONTO
{
    public int X, Y, Z;
} // fim da classe

class CS0908
{
    public static void Main ( )
    {
        // PROGRAMA PARA ACRESCENTAR COORDENADAS DE PONTOS
        // DADOS:
            // ABRIR ARQUIVO PARA GRAVAR
        FileStream ARQUIVO = new FileStream ( "PONTOS4.TXT",
            FileMode.Append );
        StreamWriter A      = new StreamWriter ( ARQUIVO );
        PONTO P = new PONTO ( );
        int X;

            // REPETIR PARA DOIS PONTOS
        Console.WriteLine ( "EXEMPLO908 - ACRESCENTAR PONTOS\n" );
        for( X = 1; X <= 2; X++ )
        {
            Console.WriteLine ( "\nENTRE COM AS COORDENADAS DE UM PONTO:" );
            Console.Write ( "\nFORNECER O VALOR DE X: " );
            P.X = int.Parse ( Console.ReadLine ( ) );
            Console.Write ( "\nFORNECER O VALOR DE Y: " );
            P.Y = int.Parse ( Console.ReadLine ( ) );
            Console.Write ( "\nFORNECER O VALOR DE Z: " );
            P.Z = int.Parse ( Console.ReadLine ( ) );

            A.Write ( P.X + " " + P.Y + " " + P.Z );
            A.WriteLine ( );
        } // FIM REPETIR
        A.Close ( ); // FECHAR ARQUIVO

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe
*/

```

```

/*
//
// ----- EXEMPLO909
//
// metodos auxiliares
using System;
using System.IO;

class PONTO
{
    public int X, Y, Z;
} // fim da classe

class CS0909
{
    public static void Main ( )
    {
        // PROGRAMA PARA PROCURAR COORDENADAS DE PONTOS
        // DADOS:
            // ABRIR ARQUIVO PARA LER
        TextReader A = new StreamReader ( "PONTOS3.TXT" );
        PONTO P = new PONTO ( );
        string LINHA;
        PONTO PROCURADO = new PONTO ( );
        bool  ACHAR;
            // REPETIR PARA DOIS PONTOS
        Console.WriteLine ( "EXEMPLO909 - PROCURAR PONTOS\n" );
        Console.WriteLine ( "\nENTRAR COM AS COORDENADAS A PROCURAR:\n" );
        PROCURADO.X = int.Parse ( Console.ReadLine ( ) );
        PROCURADO.Y = int.Parse ( Console.ReadLine ( ) );
        PROCURADO.Z = int.Parse ( Console.ReadLine ( ) );

        ACHAR = false;
        LINHA = A.ReadLine ( );
        while ( LINHA != null && ! ACHAR )
        {
            P.X = int.Parse ( LINHA );
            P.Y = int.Parse ( A.ReadLine ( ) );
            P.Z = int.Parse ( A.ReadLine ( ) );
            // SE ACHAR
            if ( P.X == PROCURADO.X && P.Y == PROCURADO.Y && P.Z == PROCURADO.Z )
            {
                ACHAR = true;
            } // FIM SE ACHAR

            LINHA = A.ReadLine ( );
        } // FIM REPETIR
        A.Close ( ); // FECHAR ARQUIVO

        if ( ACHAR )
            Console.WriteLine ( "\nCOORDENADAS ENCONTRADAS" );
        else
            Console.WriteLine ( "\nCOORDENADAS NAO ENCONTRADAS" );

        Console.Write ( "\nApertar ENTER para terminar." );
        Console.ReadLine ( );
    } // fim da acao principal
} // fim da classe

```



```

/*
/*
//
// ----- EXEMPLO910
//
// metodos auxiliares
using System;
using System.IO;

class PONTO
{
    public int X, Y, Z;
} // fim da classe

class CS0910
{
    public static void Main ( )
    {
        // PROGRAMA PARA ALTERAR COORDENADAS DE PONTOS
        // DADOS:
        PONTO P = new PONTO ( );
        FileStream ARQUIVO = null;
        byte [ ] BUFFER = new byte [ 80 ];
        int    X, BYTES;
        string LINHA;
        bool   OK = false;

        try                // ABRIR REGIAO CRITICA
        {
            // ABRIR ARQUIVO PARA LER E GRAVAR
            // SIMULTANEAMENTE
            ARQUIVO = new FileStream ( "PONTOS2.TXT",
                                     FileMode.Open,
                                     FileAccess.ReadWrite );

            OK = true;
        }
        catch ( Exception e )
        {
            Console.WriteLine ( e );
            Console.WriteLine ( "ERRO AO MANIPULAR O ARQUIVO." );
        }                // FIM DA REGIAO CRITICA

        // REPETIR PARA TODOS OS PONTOS
        if ( OK )
        {
            Console.WriteLine ( "EXEMPLO910 - ALTERAR COORDENADAS DE PONTOS\n" );

            BYTES = 0;
            while ( ( ARQUIVO.Position + 11 + 2 ) <= ARQUIVO.Length )
            {
                // LER TRES DADOS + ENTER
                Console.WriteLine ( "\nDADOS LIDOS:\n" );
                ARQUIVO.Seek ( BYTES, SeekOrigin.Begin );
                for ( X = 0; X < 13; X++ )
                {
                    BUFFER [ X ] = (byte) ARQUIVO.ReadByte ( );
                    Console.Write ( (char) BUFFER [ X ] );
                }
            } // FIM REPETIR
        }
    }
}

```

```

Console.WriteLine ( "\nQUAIS AS NOVAS COORDENADAS DO PONTO:\n" );
P.X = int.Parse ( Console.ReadLine ( ) );
P.Y = int.Parse ( Console.ReadLine ( ) );
P.Z = int.Parse ( Console.ReadLine ( ) );
    // MONTAR OS DADOS PARA SEREM GRAVADOS
LINHA = P.X.ToString ("d3") + " " +
    P.Y.ToString ("d3") + " " +
    P.Z.ToString ("d3") + "\n";
for ( X = 0; X < 11; X++ )
    BUFFER [ X ] = (byte) LINHA [ X ];
BUFFER [ 11 ] = 13;
BUFFER [ 12 ] = 10;
    // VOLTAR AO PONTO INICIAL DA LEITURA
ARQUIVO.Seek ( -13, SeekOrigin.Current );
Console.WriteLine ( "\nREGRAVAR DADOS:\n" );
for ( X = 0; X < 13; X++ )
{
    Console.Write ( (char) BUFFER [ X ] );
    ARQUIVO.WriteByte ( BUFFER [ X ] );
} // FIM REPETIR
    // AVANCAR AO PROXIMO PONTO DE LEITURA
BYTES = BYTES + 11 + 2;
} // FIM REPETIR
ARQUIVO.Close ( ); // FECHAR ARQUIVO
} // FIM SE

Console.Write ( "\nApertar ENTER para terminar." );
Console.ReadLine ( );
} // fim da acao principal
} // fim da classe
*/

```