

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace nova_cifra_de_cesar
{
    class Program
    {
        static void Main(string[] args)
        {
            while (true)
            {
                Console.Clear();
                Console.WriteLine("Digite o texto:");
                string frase = Console.ReadLine().ToUpper();
                //string frase =
                "VJNSINZJKNSINVZENSINKVOKNSINZENSINUVLKJTYNGKNNSINVJNSINNRVIVNSINQLNSINVZEWRTYNMINNS
                INNVEENSINUVININKVOKNSINZENSINGFIKLXZVJZJTYNSINXVJTYIZVSVENSINNFIUVENSINNRVIVNGKNNS
                INZTYNSINNVZJJNGKNNSINZTYNSINNVZJJNMINNSINVJNSINZJKNSINEZTYKNSINWRZINGKNNSINYVIQCZTY
                VENSINXCLVTBNLVEJTYVENSINQLDNSINXVNZEEVINNONNVONNVON";
                Console.WriteLine("Digite a chave:");
                int chave = int.Parse(Console.ReadLine());

                Console.WriteLine("1 - Criptografar");
                Console.WriteLine("2 - Descriptografar");
                Console.WriteLine("0 - Sair");
                Console.Write("Opção:");
                int op = int.Parse(Console.ReadLine());
                if (op == 1)
                {
                    frase = troca_todos_sinais_crip("", frase);
                    frase = troca_letras_crip("", frase, chave);
                    Console.WriteLine("Frase Criptografada");
                    Console.WriteLine(frase);
                    Console.ReadKey();
                }
                else if (op == 2)
                {
                    frase = troca_letras_descrip("", frase, chave);
                    frase = troca_todos_sinais_descrip("", frase);
                    Console.WriteLine("Frase Descriptografada");
                    Console.WriteLine(frase);
                    Console.ReadKey();
                }
                else if (op == 0)
                {
                    break;
                }
                else
                {
                    Console.WriteLine("Opção Inválida!");
                }
            }
        }
    }
}

```

```

    }
    static string troca_sinal(string frase_sai, string frase_ent, string sinal,
string cod_sinal)
    {
        frase_sai = frase_ent.Replace(sinal, cod_sinal);

        return(frase_sai);
    }
}

```

```

static string troca_todos_sinais_crip(string frase_sai, string frase_ent)
{
    string tts = troca_sinal(frase_sai, frase_ent, " ", "WBRW");
    tts= troca_sinal(frase_sai, tts, ",", "WVRW");
    tts = troca_sinal(frase_sai, tts, ".", "WPTW");
    tts = troca_sinal(frase_sai, tts, ";", "WPVW");
    tts = troca_sinal(frase_sai, tts, ":", "WDPW");
    tts = troca_sinal(frase_sai, tts, "!", "WEXW");
    tts = troca_sinal(frase_sai, tts, "?", "WINW");
    tts = troca_sinal(frase_sai, tts, "-", "WHFW");
}

```

```

    return (tts);
}
static string troca_todos_sinais_descrip(string frase_sai, string frase_ent)
{
    string tts = troca_sinal(frase_sai, frase_ent, "WBRW", " ");
    tts = troca_sinal(frase_sai, tts, "WVRW", ",");
    tts = troca_sinal(frase_sai, tts, "WPTW", ".");
    tts = troca_sinal(frase_sai, tts, "WPVW", ";");
    tts = troca_sinal(frase_sai, tts, "WDPW", ":");
    tts = troca_sinal(frase_sai, tts, "WEXW", "!");
    tts = troca_sinal(frase_sai, tts, "WINW", "?");
    tts = troca_sinal(frase_sai, tts, "WHFW", "-");
}

```

```

    return (tts);
}
static string troca_letras_crip(string frase_sai, string frase_ent, int k)
{
    for (int i = 0; i < frase_ent.Length; i++)
    {
        //Devolve o codigo ASCII da letra
        int asc = (int)frase_ent[i];

```

```

        //Coloca a chave fixa adicionando n posições no numero da tabela
        ASCIIi
        int ascc = asc + k;
        //subtrai 65 do numero obtido para que possa ser feito o MOD 26
        int texto_menos_65 = ascc - 65;
        // faz o resto da divisao por 26 e obtem um numero de 0 a 25
        int mod = (texto_menos_65 % 26);
        //depois de obter um numero de 0 a 25, somamos 65 para obter o
numero da letra em codigo ascii
        int texto_mais_65 = mod + 65;

```

```

        //Concatena o texto e o coloca na variável
        frase_sai += Char.ConvertFromUtf32(texto_mais_65);
    }
}

```

```

        return (frase_sai);
    }
    static string troca_letras_descrip(string frase_sai, string frase_ent, int
k)
    {
        for (int i = 0; i < frase_ent.Length; i++)
        {
            //Devolve o codigo ASCII da letra
            int asc = (int)frase_ent[i];

```

```

            //Coloca a chave fixa adicionando n posições no numero da tabela
ASCII
            int ascc = asc - k;
            //subtrai 65 do numero obtido para que possa ser feito o MOD 26
            int texto_menos_65 = ascc - 65;
            // faz o resto da divisao por 26 e obtem um numero de 0 a 25
            int mod = (texto_menos_65 % 26);
            //depois de obter um numero de 0 a 25, somamos 65 para obter o
numero da letra em codigo ascii
            int texto_mais_65 = mod + 65;
            if (texto_mais_65 < 65)
                texto_mais_65 = texto_mais_65 + 26; //entao, se o numero obtido
for menor que 65, somamos 26 ao numero e obtemos o codigo ascii do numero

```

```

            //Concatena o texto e o coloca na variável
            frase_sai += Char.ConvertFromUtf32(texto_mais_65);
        }
        return (frase_sai);
    }
}

```

```

static int procura_pal(string frase, int i, string pal)
{
    int k = -1;

```

```

    for ( int j=i;j < frase.Length; j++)
    {
        if (frase.Contains(pal))
        {
            k = j;
        }
    }
}

```

```

}

```

```

        return (k);
    }
}
}

```