

```

#include <stdio.h>
#include <cs50.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

// Solicita que ao executar o programa coloque algum argumento.
int main(int argc, string argv[])
{
    // Se o argumento for diferente de 2 palavras, sendo a primeira o
    nome do arquivo, retorna erro 1.
    if (argc != 2)
    {
        printf("Argument: ./caesar + key\n");
        return 1;
    }

    // Verifica caracter a caracter do argumento.
    for(int i = 0, n = strlen(argv[1]); i < n; i++)
    {
        // Função isdigit pergunta se o caracter olhado é ou não um
        número, se for diferente de um número, retorna erro 1.
        if(!isdigit(argv[1][i]))
        {
            printf("Argument: ./caesar + key\n");
            return 1;
        }
    }

    // Função atoi transforma um texto(string) em um inteiro(int).
    int key = atoi(argv[1]);

    // Solicita o texto a ser criptografado ao usuário.
    string text = get_string("Texto a criptografar: ");

    // Início do processo de criptografia.
    printf("Texto criptografado: ");

    // Loop para verificar caracter a caracter do texto
    for(int l = 0, d = strlen(text); l < d; l++)
    {
        // Função isalpha verifica se o caracter é uma letra( A à Z);
        // Função isupper verifica se o caracter é maiúscula.
        // É necessaria essa verificação para saber quanto temos que
        subtrair do número.
        if(isalpha(text[l]) && isupper(text[l]))
        {

```

```

        // Cálculo para andar o número de casas no alfabeto, de
acordo com a key.
        // Obs: É necessário subtrair 65 do caracter alfabetico
maiusculo, para que o "A" se torne o nº 0.
        int upperalpha = ((text[l] - 65) + key) % 26;

        // É necessário somar novamente os 65 para voltar a ser um
caracter alfabetico maiusculo.
        printf("%c", upperalpha + 65);

    }
    // Função islower verifica se o caracter é minusculo.
    else if(isalpha(text[l]) && islower(text[l]))
    {
        // É necessário subtrair 97 do caracter alfabetico minusculo,
para que o "a" se torne o nº 0.
        int loweralpha = ((text[l] - 97) + key) % 26;

        // É necessário somar novamente o 97 para voltar a ser um
caracter alfabetico minusculo.
        printf("%c", loweralpha + 97);
    }
    else
    {
        // Se não for um caracter alfabetico, imprime o mesmo sem
alteração.
        printf("%c", text[l]);
    }
}

printf("\n");
return 0;
}

```