

```

#include <stdio.h>
#include <string.h>
#include <cs50.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("Uso: ./ Chave de substituição\n");
        return 1;
    }

    int ndigit = strlen(argv[1]);
    int argupper[ndigit], arglower[ndigit];

    // Verifica caractere a caractere do argumento.
    for (int i = 0; i < ndigit; i++)
    {
        // Função isdigit pergunta se o caractere olhado é ou não um
        número, se for um número, retorna erro 1.
        if (isdigit(argv[1][i]))
        {
            printf("Uso: ./ Chave de substituição\n");
            return 1;
        }
    }

    // Verifica se a key tem menos ou mais de 26 dígitos, caso tenha,
    retorna o erro 1.
    if (ndigit < 26)
    {
        printf("A chave deve conter 26 caracteres.\n");
        return 1;
    }
    else if (ndigit > 26)
    {
        printf("A chave deve conter 26 caracteres.\n");
        return 1;
    }

    // Verifica caractere a caractere a fim de arrumar a key, separando
    em 2, uma de letras maiúsculas e uma de minúsculas.
    for (int i = 0; i < ndigit; i++)
    {

```

```

        // Se a letra for maiúscula salva normal na variável "argupper" e
        altera a letra maiúscula para minúscula e salva na "arglower".
        if (isupper(argv[1][i]))
        {
            argupper[i] = argv[1][i];
            arglower[i] = argv[1][i] + 32;
        }
        // Executa o inverso do if acima.
        else if (islower(argv[1][i]))
        {
            arglower[i] = argv[1][i];
            argupper[i] = argv[1][i] - 32;
        }
    }

    // Verifica se tem caracteres repetidos na key.
    for (int i = 0; i < ndigit; i++)
    {
        for (int x = 0; x < ndigit; x++)
        {
            if (i != x && argupper[i] == argupper[x])
            {
                printf("A chave não pode conter caracteres repetidos\n");
                return 1;
            }
        }
    }
    // ENCERRA A VERIFICAÇÃO SE A KEY É VÁLIDA!

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

    int ntext = strlen(text);

    // Inicio da criptografia.
    printf("ciphertext: ");

    // Olha caractere a caractere do texto a ser criptografado.
    for (int i = 0; i < ntext; i++)
    {
        // Se for uma letra do alfabeto maiúscula.
        if (isalpha(text[i]) && isupper(text[i]))
        {
            // Pega a letra e altera pela letra na posição correspondente
            na key maiúscula ("argupper").
            text[i] = argupper[text[i] - 65];
            printf("%c", text[i]);
        }
    }

```

```
    }  
    // Se for uma letra do alfabeto minúscula.  
    else if (isalpha(text[i]) && islower(text[i]))  
    {  
        // Pega a letra e a altera pela letra na posição  
correspondente na key minúscula ("arglower").  
        text[i] = arglower[text[i] - 97];  
        printf("%c", text[i]);  
    }  
    // Caso seja um caractere diferente, o imprime sem altera-lo.  
    else  
    {  
        printf("%c", text[i]);  
    }  
}  
  
printf("\n");  
return 0;  
}
```