

Este é o CS50



Introdução à Ciência da Computação (CS50)


OpenCourseWare

Doar  (<https://cs50.harvard.edu/donate>)

David J. Malan (<https://cs.harvard.edu/malan/>)

malan@harvard.edu

 (<https://www.facebook.com/dmalan>)  (<https://github.com/dmalan>) 

(<https://www.instagram.com/davidjmalan/>)  (<https://www.linkedin.com/in/malan/>)

 (<https://www.reddit.com/user/davidjmalan>) 

(<https://www.threads.net/@davidjmalan>)  (<https://twitter.com/davidjmalan>)

Substituição

Problema a resolver

Em uma cifra de substituição, "criptografamos" (ou seja, ocultamos de forma reversível) uma mensagem substituindo cada letra por outra. Para isso, usamos uma *chave*: neste caso, um mapeamento de cada letra do alfabeto para a letra correspondente na criptografia. Para "descriptografar" a mensagem, o destinatário precisa conhecer a chave para reverter o processo: traduzir o texto criptografado (geralmente chamado de *texto cifrado*) de volta para a mensagem original (geralmente chamada de *texto plano*).

Uma chave, por exemplo, pode ser a string `NQXPOMAFTRHLZGECYJIUWSKDVB`. Esta chave de 26 caracteres significa que `A` (a primeira letra do alfabeto) deve ser convertida em `N` (o primeiro caractere da chave), `B` (a segunda letra do alfabeto) deve ser convertida em `Q` (o segundo caractere da chave) e assim por diante.

Uma mensagem como `HELLO`, então, seria criptografada como `FOLLE`, substituindo cada uma das letras de acordo com o mapeamento determinado pela chave.

Em um arquivo chamado `<nome_do_arquivo>_substitution.c` em uma pasta chamada `<nome_da_pasta>_substitution`, crie um programa que permita criptografar mensagens usando uma cifra de substituição. No momento da execução do programa, o usuário deverá escolher, por meio de um argumento de linha de comando, qual será a chave da mensagem secreta que será fornecida em tempo de execução.

Demonstração

```
$ ./substitution
Usage: ./substitution key
$ ./substitution ABC
Key must contain 26 characters.
$ ./substitution 1 2 3
Usage: ./substitution key
$ ./substitution YTNSHKVEFXRBAUQZCLWDMIPGJO
plaintext:  Hello!
ciphertext: Ehbbq!
$
```

Recorded with **asciinema**

Especificação

Projete e implemente um programa, `substitution`, que criptografe mensagens usando uma cifra de substituição.

- Implemente seu programa em um arquivo chamado `substitution.c` em um diretório chamado `substitution`.
- Seu programa deve aceitar um único argumento de linha de comando, a chave a ser usada para a substituição. A chave em si não deve diferenciar maiúsculas de minúsculas, portanto, o fato de qualquer caractere na chave ser maiúsculo ou minúsculo não deve afetar o comportamento do seu programa.
- Se o seu programa for executado sem nenhum argumento de linha de comando ou com mais de um argumento de linha de comando, ele deverá imprimir uma mensagem de erro de sua escolha (com `printf`) e retornar imediatamente `main` um valor de `1` (que geralmente indica um erro).
- Se a chave for inválida (por exemplo, por não conter 26 caracteres, por conter qualquer caractere que não seja alfabético ou por não conter cada letra exatamente uma vez), seu programa deverá imprimir uma mensagem de erro de sua escolha (com `printf`) e retornar imediatamente com `main` o valor `1`.

- Seu programa deve exibir `plaintext:` (sem nova linha) e, em seguida, solicitar ao usuário uma `string` entrada de texto simples (usando `get_string`).
- Seu programa deve exibir `ciphertext:` (sem quebra de linha) o texto original seguido do texto cifrado correspondente, com cada caractere alfabético do texto original substituído pelo caractere correspondente no texto cifrado; caracteres não alfabéticos devem ser exibidos sem alterações.
- Seu programa deve preservar a distinção entre maiúsculas e minúsculas: letras maiúsculas devem permanecer maiúsculas; letras minúsculas devem permanecer minúsculas.
- Após imprimir o texto cifrado, você deve imprimir uma nova linha. Seu programa deve então terminar retornando `0` da função `main`.

Você pode achar uma ou mais funções declaradas `ctype.h` úteis, conforme manual.cs50.io (<https://manual.cs50.io>).

Passo a passo



Como testar

Correção

No seu terminal, execute o comando abaixo para verificar se o seu trabalho está correto.

```
check50 cs50/problems/2025/x/substitution
```

▼ Como usar **debug50**

Deseja executar o comando `debug50`? Você pode fazê-lo da seguinte forma, após compilar seu código com sucesso usando o comando `make`:

```
debug50 ./substitution KEY
```

onde `<key>` `KEY` é a chave que você fornece como argumento de linha de comando para o seu programa. Observe que a execução

```
debug50 ./substitution
```

(Idealmente!) fará com que seu programa termine solicitando uma chave ao usuário.

Estilo

Execute o comando abaixo para avaliar o estilo do seu código usando `style50`.

```
style50 substitution.c
```

Como enviar

No seu terminal, execute o comando abaixo para submeter seu trabalho.

```
submit50 cs50/problems/2025/x/substitution
```