# This is CS50x

OpenCourseWare

Donate (https://cs50.harvard.edu/donate)

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

f (https://www.facebook.com/dmalan) (https://github.com/dmalan) (https://www.instagram.com/davidjmalan/) (https://www.linkedin.com/in/malan/)

(https://orcid.org/0000-0001-5338-2522) **Q** 

(https://www.quora.com/profile/David-J-Malan)

(https://www.reddit.com/user/davidjmalan) \*\* (https://twitter.com/davidjmalan)

## **Substitution**

Implement a program that implements a substitution cipher, per the below.

\$ ./substitution JTREKYAVOGDXPSNCUIZLFBMWHQ

plaintext: HELLO
ciphertext: VKXXN

### **Background**

In a substitution cipher, we "encrypt" (i.e., conceal in a reversible way) a message by replacing every letter with another letter. To do so, we use a *key*: in this case, a mapping of each of the letters of the alphabet to the letter it should correspond to when we encrypt it. To "decrypt" the message, the receiver of the message would need to know the key, so that they can reverse the process: translating the encrypt text (generally called *ciphertext*) back into the original message (generally called *plaintext*).

A key, for example, might be the string NQXPOMAFTRHLZGECYJIUWSKDVB. This 26-character key means that A (the first letter of the alphabet) should be converted into N (the first character of the key), B (the second letter of the alphabet) should be converted into Q (the second character of the key), and so forth.

A message like HELLO, then, would be encrypted as FOLLE, replacing each of the letters according to the mapping determined by the key.

Let's write a program called substitution that enables you to encrypt messages using a substitution cipher. At the time the user executes the program, they should decide, by providing a command-line argument, on what the key should be in the secret message they'll provide at runtime.

Here are a few examples of how the program might work. For example, if the user inputs a key of YTNSHKVEFXRBAUQZCLWDMIPGJO and a plaintext of HELLO:

```
$ ./substitution YTNSHKVEFXRBAUQZCLWDMIPGJO
plaintext: HELLO
ciphertext: EHBBQ
```

Here's how the program might work if the user provides a key of VCHPRZGJNTLSKFBDQWAXEUYMOI and a plaintext of hello, world:

```
$ ./substitution VCHPRZGJNTLSKFBDQWAXEUYMOI
plaintext: hello, world
ciphertext: jrssb, ybwsp
```

Notice that neither the comma nor the space were substituted by the cipher. Only substitute alphabetical characters! Notice, too, that the case of the original message has been preserved. Lowercase letters remain lowercase, and uppercase letters remain uppercase.

Whether the characters in the key itself are uppercase or lowercase doesn't matter. A key of VCHPRZGJNTLSKFBDQWAXEUYMOI is functionally identical to a key of vchprzgjntlskfbdqwaxeuymoi (as is, for that matter, VcHpRzGjNtLsKfBdQwAxEuYmOi).

And what if a user doesn't provide a valid key?

```
$ ./substitution ABC
Key must contain 26 characters.
```

Or really doesn't cooperate?

```
$ ./substitution
Usage: ./substitution key
```

Or even...

```
$ ./substitution 1 2 3
Usage: ./substitution key
```

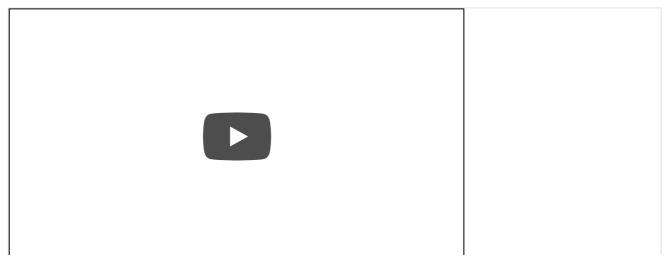
#### ► Try It

### Specification

Design and implement a program, substitution, that encrypts messages using a substitution cipher.

- Implement your program in a file called substitution.c in a directory called substitution.
- Your program must accept a single command-line argument, the key to use for the substitution. The key itself should be case-insensitive, so whether any character in the key is uppercase or lowercase should not affect the behavior of your program.
- If your program is executed without any command-line arguments or with more than one command-line argument, your program should print an error message of your choice (with printf) and return from main a value of 1 (which tends to signify an error) immediately.
- If the key is invalid (as by not containing 26 characters, containing any character that is not an alphabetic character, or not containing each letter exactly once), your program
  - should print an error message of your choice (with printf) and return from main a value of 1 immediately.
- Your program must output plaintext: (without a newline) and then prompt the user for a string of plaintext (using get\_string).
- Your program must output ciphertext: (without a newline) followed by the plaintext's corresponding ciphertext, with each alphabetical character in the plaintext substituted for the corresponding character in the ciphertext; non-alphabetical characters should be outputted unchanged.
- Your program must preserve case: capitalized letters must remain capitalized letters; lowercase letters must remain lowercase letters.
- After outputting ciphertext, you should print a newline. Your program should then exit by returning from main.

### Walkthrough



I, 7:33 PM	Substitution - CS50x 2021
How to Test Your Cod	de
Execute the below to evaluate compile and test it yourself a	te the correctness of your code using check50. But be sure to as well!
check50 cs50/problems/20	021/x/substitution
Execute the below to evaluat	te the style of your code using style50.
style50 substitution.c	
How to Submit	
	n with your GitHub username and password when prompted. F  *) instead of the actual characters in your password.
submit50 cs50/problems/2	1921/x/substitution