Practice S08P07: InfoSec and Cryptography I: ROT-13

http://www.comp.nus.edu.sg/~cs1010/4_misc/practice.html

Week of release: Week 9

Objective: Characters and Strings

Task statement:

As more and more information is stored in computers and transmitted over network, it is critical to prevent unauthorized parties from accessing such information.

A common approach for this purpose is to encrypt such information into ciphertext so that it cannot be easily understood by unauthorized parties. In contrast, the authorized parties are informed of how to decrypt the cipher text. Therefore, they will be able to access the information without any problem.

For example, an encryption algorithm called ROT-13 works by substituting a letter in a message with another letter which is 13 places further along the English alphabet, wrapping back to the beginning if necessary. For example, A (1st letter in the English alphabet) is replaced by N (14th), while W (23rd) is replaced by J (10th). The complete rules for the algorithm are as shown in Fig. 1.

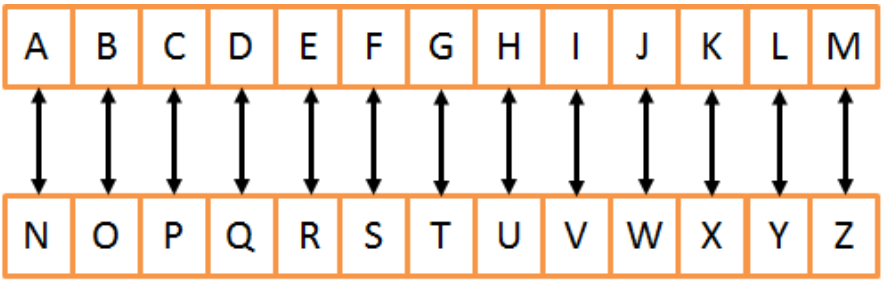

Figure 1. The complete substitution rules for the ROT-13 algorithm.

Using this algorithm, the message "HELLO" is encrypted into the ciphertext "URYYB". In addition, by reversing the steps in the algorithm, the cipher text "URYYB" can be decrypted back to into the message "HELLO".

Write a program **rot.c** to read in an integer (1 or 2) and a string (a sequence of uppercase and / or lowercase letters in the English alphabet):

- If the given integer is 1, the program encrypts the string using the ROT-13 algorithm.
- If the given integer is 2, the program decrypts the string by reversing the ROT-13 algorithm.

Your program should contain two functions with the following headers:

```
void encrypt(char message[], char result[])
void decrypt(char message[], char result[])
```

where **`message`** is the given string and **`result`** is a char array for storing the result of encrypting / decrypting the message.

You are only allowed to declare and use two char arrays, one for storing the given string and another for storing the results. You may assume that the message contains at most 80 letters.

You may write additional function(s) if necessary. You may use any characters and string function(s) if necessary.

Sample runs

```
Enter 1 for encryption, 2 for decryption: 1
Enter message: beef
Encrypted message: orrs

Enter 1 for encryption, 2 for decryption: 2
Enter message: UVTU
Decrypted message: HIGH

Enter 1 for encryption, 2 for decryption: 1
Enter message: heLLo
Encrypted message: uyYYb
```