Software Systems

Mini Assignment #3

Due: October 22, 2017 on myCourses at 23:30

Cryptography in C

In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.

Example: The transformation can be represented by aligning two alphabets; the cipher alphabet is the plain alphabet rotated left or right by some number of positions. For instance, here is a Caesar cipher using a left rotation of three places, equivalent to a right shift of 23 (the shift parameter is used as the key):

```
Plain:    ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:   XYZABCDEFGHIJKLMNOPQRSTUVW
```

When encrypting, a person looks up each letter of the message in the "plain" line and writes down the corresponding letter in the "cipher" line.

```
Plaintext:  THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Ciphertext: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD
```

Deciphering is done in reverse, with a right shift of 3.

Notice that modulo is helpful, assuming that, A=0, B=1, C=3, …, Z=25. Then to shift characters one needs to do the following:

> Int letter = ‘A’, key = 3, newLetter;
>
> newLetter = (letter + key) % 26; // since 26 letters in the alphabet
>
> Note that this does not work in the opposite direction since subtracting 3 could lead to a negative number.

Notice further, we can convert ASCII to A=0, B=1, C=3, etc. by performing this subtraction: int letter; letter = letter – ‘A’; for upper case characters. You would need to subtract by ‘a’ for lower case characters. This would work for the entire alphabet.

This mini assignment is asking you to write a C language program to implement this cipher.

The program takes two inputs:

1) A sentence typed in by the user at a prompt.
2) The key the user plans to use (For the assignment, we will restrict it to a left shift between 1 and 25. Ex, a 3 means shift left by 3 places), typed in at a prompt.

The program displays:

1) the original sentence and
2) the encrypted sentence.
3) It then decrypts the encrypted message and displays the decrypted message. The decrypted message and the original message should agree.

The input sentence will only be text using alphabets with spaces as in the above example. We will not test it for numbers or special characters.

Expected format:

```
Bash-prompt $ ./a.out
Sentence: Mary
Key: 2
Original message: Mary
Encrypted message: Kypw
Decrypted message: Mary
Bash-prompt $
```

HOW IT WILL BE GRADED

Points removed for bad practices:

- -1 for not following instructions
- -1 for not indenting, spacing, and/or commenting
- -1 for not using good variable names

This assignment is worth 20 points:

- +2 Input sentence stored in an array and key in variable
- +2 Output as described above
- +4 Arrays, proper use of
- +2 Modulo, proper use of
- +5 Encryption, proper functioning of, for all letters in sentence
- +5 Decryption, proper functioning of, for all letters in sentence