ECE 175: Computer Programming for Engineering Applications

Homework Assignment 4

Conventions: Name your *C* programs as *hwx_py.c* where *x* corresponds to the homework number and *y* corresponds to the problem number. As an example the *C* program for hw1 problem 1 should be named as *hw1_p1.c*.

Submission Instructions: Use the dropbox on D2L to submit the .c files and a PDF of your pseudocode

1 Caesar's Cipher (70 points)

The *Caesar's cipher* is a cryptographic method for encrypting text such that it becomes unreadable to a party without access to the cryptographic key. It is named after Julius Caesar, who allegedly used it to protect messages of military significance. The encryption and decryption operations are simple shifts of the alphabet letters in a cyclic fashion. Mathematically, for a key k, the encryption and decryption are defined as:

Encryption: $c = (x + k) \mod 26$,

Decryption: $x = (c - k) \mod 26$.

To illustrate the use of the Ceasar's cipher, consider the encryption of "fry" with key $k = 3$. Letters *f, r, y* correspond to the 5th, 16th, and 24th letter of the alphabet (letter *a* being the 0th).

$$\begin{aligned} \text{f: } (5+3) \mod 26 &= 8 \rightarrow i \\ \text{r: } (16+3) \mod 26 &= 19 \rightarrow u \\ \text{y: } (24+3) \mod 26 &= 1 \rightarrow b \end{aligned} \tag{1}$$

Similarly, the decryption of *i,u,b* follows the reverse process

$$\begin{aligned} \text{i: } (8-3) \mod 26 &= 5 \rightarrow f \\ \text{u: } (19-3) \mod 26 &= 16 \rightarrow r \\ \text{b: } (1-3) \mod 26 &= 24 \rightarrow y \end{aligned} \tag{2}$$

NOTE: The modulo operation for negative numbers is different from the % arithmetic operator in *C*.

Write a *C* program that decrypts a file named "encrypted.txt" and places the decryption output to a file called "decrypted.txt". A file encrypted with $k = 3$ is provided with the assignment. You can use it to test your decryption function. You will know when you have succeeded because the text becomes readable. In your program

- Ask the user to enter the decryption key.
- Repeat the request until the right key is entered.
- Display a message when file decryption is over.
- Ask the user to enter a key to re-encrypt the file.
- Re-encrypt file "decrypted.txt" and store it at "encrypted.txt"

- Display a message when file encryption is over.

Only alphabet letters (uppercase/lowercase) must be encrypted. The remaining characters (question marks, periods, etc. must remain intact).

Your code must be modular. Use the following function prototypes for encryption and decryption:

char decrypt_fun(int shift, char letter); // This function receives the shift value (key) and an encrypted letter and returns the decrypted letter.

char encrypt_fun(int shift, char letter); // This function receives the shift value (key) and the plaintext letter and returns the encrypted letter.

You are also given the following function that correctly implements the modulo operation for both positive and negative numbers

```
int mod (int x, int y){
   while(x < 0){
     x+ = y;
   }
return x%y;
}
```

Sample Code Execution: **Red text indicates information entered by the user**

```
Enter the decryption key:5
Wrong key
Enter the decryption key:3
File decrypted!
Do you want to re-encrypt the file (y/n)? y
Enter an encryption key: 6
File encrypted!
Goodbye
```

2 Lab 4 Assignment (30 points)

Submit a PDF of your pseudocode and the corresponding .c file named hw4_p1.c, via D2L dropbox