

# CIS\*2500 Lab Assignment 3

## Creating a Caesar Cipher

Read about Caesar Ciphers from the wikipedia [http://en.wikipedia.org/wiki/Caesar\\_cipher](http://en.wikipedia.org/wiki/Caesar_cipher).

### L3.1 (5 marks)

Create a program called **copy\_text** that takes two command line arguments: the first argument is the name of the file to be copied; the second argument is the name of the new file. If the second argument is missing, copy the file to stdout. If both arguments are missing, the program should read from stdin and print to stdout.

Example:

```
copy_text original.txt theCopy.txt
```

### L3.2 (5 marks)

The most common of the Caesar Ciphers still in use is Rot13, which uses a shift of 13. Write a program that takes in input from a file and prints it into a new file encoded using Rot13. Only Rot13 letters, leave digits, white space and punctuation unchanged. The file names should be command line arguments as specified for the `copy_text` program from Question 1.

Name this program **rot13.c**

### L3.3 (10 marks)

Encode a message from a file using the Caesar Cipher with a shift as entered as a command line argument. The results should be placed in a new file. If the shift is 0, the program should just produce a copy of the file (similar behavior as question 1). The program should accept positive and negative shift values (positive shifts add the absolute shift amount; negative shifts subtract the absolute shift amount). If a command line argument is entered that is not a number, or is a number whose absolute value is greater than or equal to 26, print an appropriate error message to the screen; but do not create a new file (use `fprintf(stderr, "%s\n", "your message")` to accomplish this).

The program should take in a three command line arguments:

- the shift number
  - an integer between -25 and 25 inclusive
- -F input\_file\_name
  - the name of the input file

- there is usually a space between -F and the file name
    - but there doesn't have to be - you have to handle both cases
- -O output\_file\_name
  - the name of the output file
  - again, there is usually a space between -O and the file name
    - but there doesn't have to be - you have to handle both cases

The shift number does not have a option name; it is just a number

- If the -O argument is missing, output should go to stdout.
- If the -F argument is missing, the program should read from stdin.
- If no shift number is present, the program should use a Rot13 encoding.

Name this program **encode.c** If you have this program working, you do not need to hand in rot13.c, which will be marked by running encode.c with no shift number.

## NOTES (IMPORTANT INFO ... READ CAREFULLY)

1. Your code must compile cleanly with no error or warning messages using the -Wall flags in gcc.
2. Your source code should be properly formatted and meaningful variable names should be used.
3. Your source code should contain brief comments describing the functionality and the major components of each procedure. Any complex structures should also be commented.
4. If you hand in an assignment that does not compile you will get a mark of zero.
5. All work in this course is to be done independently.
6. This Lab will be graded in person during your scheduled lab grading period. If you are not graded in person by one of your assigned TAs (or TA scheduled along side them), you will receive a 0.
7. You should also submit your source code file (.c and .h files) and a makefile to the L3 submission link in the course website. Note: this is not a substitute for in person grading.