

# Lab 7 – Caesar Cipher

75 possible points

## Goal:

Since you are now familiar with sub routines (procedures) lets use them to create a cipher.

## Encryption and Decryption

You want to be able to hang out with the cool kids that use the Caesar Cipher ( [http://en.wikipedia.org/wiki/Caesar\\_cipher](http://en.wikipedia.org/wiki/Caesar_cipher) ) to tell jokes on the internet so you decide to write a program to help you convert jokes into and out of any Caesar Cipher format (ROT-13 is an example of one). Your task is:

- Create a 2x200 array. You will be able to store one string in normal text mode and the encrypted string for it.
- Print a prompt that asks to either encrypt or decrypt a string.
- Print a prompt that ask the user for a number between 1 and 25 or 'x' to quit.
- Print a prompt that asks for the string to encrypt or decrypt.
- Using either row major or column major form store the string entered into your 2D array in row 0.
- Either encrypt or decrypt the string entered and store it into your 2D array in row 1.
- Print out your 2D array.
- Repeat.

You need to also make a flowchart for this program.

## The Caesar Cipher Algorithm

The Caesar Cipher is an encoding method that works like this (example is for a shift amount of 13, which is a ROT-13).

```
if ( character is in alphabet )
{
    if ( character is in top half of alphabet [A-M|a-m] )
        character = character + 13;
    else ( character is in bottom half [N-Z|n-z] )
        character = character - 13;
}
print (character);
```

## Notes:

- Non-alphabetic characters such as numbers (1,2,3), punctuation (, . ? !), and other weird stuff (& # \$ % ' \ | , etc) should not be altered before output. That is to say, if I enter '@', I should get '@' back.
- To get some test input do an internet search for “rot13 jokes”, this will be for Caesar Cipher’s of 13 only though.

- You can also search the internet for other Caesar Cipher examples.

## Requirements

You are required to use a 2D array structure with either **Row Major** or **Column Major** ordering that was covered in class. Be sure to put which one you use in your README and in your code comments. You need to make use of a few sub routines:

- Encrypt: It takes a character and a cipher as an input and returns the encrypted value
- Decrypt: It takes an encrypted character and a cipher and returns the decrypted value
- Print Array: It prints the 2D array out
- Store: Takes coordinates (Ri, Ci) and byte of data and stores into the 2D array
- Load: Takes coordinates (Ri, Ci) and loads a byte of data from the 2D array

**The Encrypt, Decrypt, and Print Array sub routines will call the Store and Load sub routines.**

## Example of possible output for Part B:

```
Hello, welcome to my Caesar Cipher program
Do you want to (E)ncrypt or D(ecrypt) or e(X)it?
> D
What is the cipher (1-25)?
> 13
What is the string (up to 200 characters)?
> Fpubby vf sha!!
Here is your string and the decrypted result
<Encrypted> Fpubby vf sha!!
<Decrypted> School is fun!!
> Do you want to (E)ncrypt or D(ecrypt) or e(x)it?
> X
Goodbye!!
```

See the grading template for information on the points possible for the extra credit.

## Files to Submit:

- Lab7.asm
- Flow chart shown to TA/Tutor
- Lab report

## Check-off:

You must demonstrate this lab to the TA/tutor, either before submission or after submission. To be considered on time you must have submitted everything by the due date.

## Files to Submit:

- Lab7.asm

- Flow chart shown to TA/Tutor
- Lab report

**Check-off:**

You must demonstrate this lab to the TA/tutor, either before submission or after submission. To be considered on time you must have submitted everything by the due date.