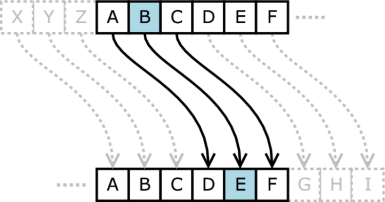
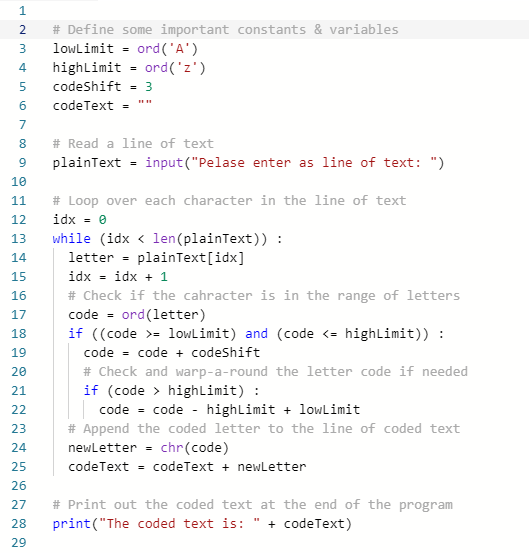
**Background:**

You will be working with a partner to write a pair of programs that will secretly encode a text file and then decode the file back to plain text.

In a previous lesson you have already implemented a simple   
shift code (Caesar Cipher) that added 3 to the ASCII code of a letter.   
The Caesar cipher is one of the earliest known and simplest ciphers.   
It is a type of substitution cipher in which each letter in the plaintext   
is 'shifted' a certain number of places down the alphabet. If the shift   
is greater than ‘z’ then the shift wraps around to “a’.

The sample program below does the following to implement the Caesar Cipher:

1. Reads a line of text from the console input
2. Loops through each character in the line of text
3. Checks that the character is a printable letter
4. Shifts the letter by 3 positions (wrapping back to “a” if necessary)
5. Appends the shifted letter a new line of coded text
6. At the end of the program, prints the line of coded text to the console display.



**Assignment:**

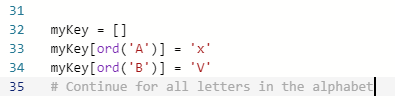
1. Work with a partner to create a Substitution Cipher Key (See the assignment slide notes for more information)
2. Partner #1 will write a program to do the following:
   1. Open a text file called “MessageText.txt” for reading
   2. Open a text file called “CodedText.txt” for writing
   3. Loop over each line of text in the message file
   4. Code each line of text using the Substitution Cipher Key
   5. Write each line of coded text to the code file
   6. Close both the message file and code file
3. Partner #2 will write a program to do the following:
   1. Open the text file called “CodedText.txt” for reading
   2. Open a text file called “DecodedText.txt” for writing
   3. Loop over each line of text in the coded text file
   4. De-Code each line of text using the Substitution Cipher Key
   5. Write each line of de-coded text to the de-code file
   6. Close both the code file and de-code file
4. Your programs should work for messages that contain multiple lines of text.
5. Be prepared to demonstrate and explain your coding / de-coding program and code files.

**Programming Hints:**

Use the sample Caesar Cipher program as a starting point

Get you program to work with single lines of text from console input / output before adding the use of files. (It is easier to debug console input / output than file input / output.)

Implement your key using the list data type. An example code is as follows:



# Define some important constants & variables

lowLimit = ord('A')

highLimit = ord('z')

codeShift = 6

codeText = ""

# Read a line of text

plainText = input("Please enter as line of text: ")

# Loop over each character in the line of codeText

idx = 0

while (idx < len(plainText)) :

letter = plainText[idx]

idx = idx + 1

#Check if the character is in the range of letters

code = ord(letter)

if ((code >= lowLimit) and (code <= highLimit)) :

code = code + codeShift

# Check and warp-a-round the letter code if needed

if (code > highLimit) :

code = code - highLimit + lowLimit

# Append the coded letter to the line of coded text

newLetter = chr(code)

codeText = codeText + newLetter

# Print out the coded text at the end of the program

print("The coded text is: " + codeText)

myKey = []

myKey[ord('A')] = 'G'

myKey[ord('B')] = 'J'

myKey[ord('C')] = 'A'

myKey[ord('D')] = 'P'

myKey[ord('E')] = 'N'

myKey[ord('F')] = 'S'

myKey[ord('G')] = 'H'

myKey[ord('H')] = '3'

myKey[ord('I')] = 'M'

myKey[ord('J')] = '5'

myKey[ord('K')] = 'R'

myKey[ord('L')] = '6'

myKey[ord('M')] = 'E'

myKey[ord('N')] = 'B'

myKey[ord('O')] = '2'

myKey[ord('P')] = '7'

myKey[ord('Q')] = 'D'

myKey[ord('R')] = 'L'

myKey[ord('S')] = '8'

myKey[ord('T')] = 'X'

myKey[ord('U')] = 'C'

myKey[ord('V')] = '1'

myKey[ord('W')] = '4'

myKey[ord('X')] = 'U'

myKey[ord('Y')] = '9'

fileHandle = open("newfile.txt","w+")

fileHandle.write("Hello, this is a new file.\n")

fileHandle.write("You should see this text when you")

fileHandle.write("select the file in the file chooser window.\n")

fileHandle.close()