COSC 251 – Programming Languages
Project 4
Spring 2017

Objective: Vkxtmx t lxkbxl hy vbiaxk yngvmbhgl bg EBLI.

Your Task:

For this project, you will implement three classic ciphers in LISP. Each cipher function will take an input as parameters and return the encoded message using the cipher methodology.

Caesar Cipher – also known as the shift cipher, this is one of the oldest ciphers in cryptography and was supposedly used by Julius Caesar in his private correspondence. In order to calculate this cipher you take the string **s** and shift (or rotate) each letter (allowing for the alphabet to wrap back to A) in **s** by a certain number **n**. If **n == 4**, then each letter is shifted by 4 positions. So, an A becomes an E, and a Y becomes a C.

Signature: (caesar s n) ;s is our string to be encoded, n is our rotation integer
Example: (caesar “COSC is awesome” 6) will output “IUYI oy gckyusk”

Vigenère Cipher – created by the Italian cryptographer Giovan Battista Bellaso in 1553 and then misattributed to the French cryptographer Blaise de Vigenère, this cipher uses a similar algorithm to the Caesar cipher, except instead of a single number key, the key is a string **s2**. The algorithm I’d like you to use proceeds as follows: for each letter in our message string **s1**, we shift the letter by the ASCII value of a character in the key assigned based on position (if the key is shorter than our message, then repeat the key). If the resulting ASCII value would be outside of the alphabet, then wrap around back to “a”. Note that we only use the lower case letters for our cipher. For instance, if **s1** is “lindsay” and **s2** is “abc” then we calculate our values:

s1	**s2**	**result**
l	a	108 + 97 = e
i	b	105 + 98 = c
n	c	110 + 99 = i
d	a	100 + 97 = w
s	b	115 + 98 = m
a	c	97 + 99 = v
y	a	121 + 97 = r

Signature: (vig s1 s2) ;s1 is our string to be encoded, s2 is our key
Example: (vig “lindsay” “abc”) will output “eciwmvr”

ADFGVX Cipher – used by the Germans in World War 1, this is a fantastic cipher that was very difficult to break. For our algorithm, you will have three parameters: **s**, the message; **k** a keyword; and then **sq**, the key square (6x6) for encoding, written as a string of 36 characters. You will preserve any spaces in the message, but not use them during the column transposition.

For the algorithm follow the procedure found here:

http://practicalcryptography.com/ciphers/classical-era/adfgvx/

Signature: (adf s k sq) ;s is our string to be encoded, k is our keyword, sq is our key square as a string
Example: (adf "lisp is awful" "stmary" "ph0qg64mea1yl2nofdxkr3cvs5zw7bj9uti8") will output
"VVGAXXVF VVFF ADXAAGFAAV"

You do not need to verify that the parameters are in the right format or type.

Restrictions: you may not use the loop macro, while loop function, do loop function, or any function that creates a loop-like control structure. You may not use any third party libraries. The message can be any length and can contain upper and lower case letters in the Caesar cipher, only lower case letters in the Vigenère cipher, and any alphanumeric character in the ADFGVX cipher. All messages can contain space characters and those spaces should be preserved. The keywords **s2** in the Vigenère cipher and **k** in the ADFGVX cipher can also be any length, but will not contain any non-letter characters, including whitespace.

Implementation details: you must implement all three ciphers in a single file, called Proj4.lisp. You must match the signatures of the cipher functions as listed above, with the proper number of parameters. You will be provided a driver Soon™ and it will have instructions for execution and all test cases that will be run against your code. Your solution must work with my driver.

You may work in groups of three on this project. Teams must be sent to me via email by 5pm April 6th. Putting in a request past this point will be accepted but there will be an automatic deduction of 5 points per day.

Expectations: The code should be clean, concise, well-commented and correct. If you use an outside source, be sure to document that source. Significant use of outside sources will result in a deduction. Grading rubric will be provided a week ahead of the due date.

DUE: April 21st, 11:59pm via Blackboard.

Credit: University of Mary Washington for the Vigenère cipher problem.