

# Project 5

CS 261

Save your program for this project as `<emailID>_project5.py` where `<emailID>` is the part of your Hampden-Sydney e-mail address before the `@` symbol. When you are finished, e-mail your program to `blins@hsc.edu`. Your solution is due by noon on Friday, October 11.

## Caesar Cipher

The Caesar cipher is one of the oldest and simplest methods for encrypting text. You pick a shift amount and then you shift every letter in the message by that amount. For example, if the shift is 2, then a's become c's, b's become d's and so on, until y and z wrap around to become a and b, respectively. A message like:

about to get hazy

would become:

cdqvw vq igv jcba.

For this project, you will write three functions related to the Caesar cipher. You may also include any additional functions that are helpful to construct the three functions described below. You should not import any libraries for this project.

1. Write a function called `caesar_shift(message, shift)` that applies a Caesar shift to any string. You can assume that the message will not have any upper case letters, and you only need to shift letters, not any other symbols.
2. The Caesar cipher is not a good way to encrypt a message because it is very easy to decrypt. In fact, you can often figure out the shift by finding the most common letter in the encrypted text. Since "e" is the most common letter in most English texts, there is a good chance that the shift is just the amount needed for "e" to become the most common letter in your encrypted message. Write a function called `letter_frequency(message)` that returns a list. The value of the output list at index  $i$  should be the number of times the  $i$ -th letter of the alphabet occurs in the message. So the first entry of the output is the number of a's, then b's, and so on.
3. In order to find the letter that occurs the most in the message, we need a way to search through the output of the letter frequency list and find the index with the largest entry. Write a function called `argmax(list)` that will search a list of numbers and return the *index* of the largest entry in the list. If multiple entries in the list are tied for the largest value, then return the index of the first one that is equal to the maximum.

## Hints

1. It would help to have functions to convert the letters (a, b, c, ..., z) to the numbers (0, 1, 2, ..., 25) and vice versa:

"a"  $\rightarrow$  0

"b"  $\rightarrow$  1

"c" → 2  
⋮  
"z" → 25

There are at least two ways to do this. One option is to create a string with all of the letters of the alphabet in order and use the `.index()` method. Another option is to use the built in functions `ord()` and `chr()`. The function `ord()` converts any character to its integer ASCII value. So `ord("a")` returns 97, `ord("b")` returns 98, and so on. The function `chr()` does the reverse, it converts an integer to its corresponding ASCII character. So `chr(97)` is "a", `chr(98)` is "b", etc.

2. Don't forget methods like `.count()` when working with lists and strings.
3. For the `argmax()` function, you'll need to loop over every value in the input list. Make sure to use an accumulator variable to keep track of which index corresponds to the maximum value you have seen so far.