

# Simple Substitution Ciphers

---

## Introduction

In this exercise, we will complete several pieces of Python programs to encode and decode messages by using simple substitution ciphers.

## Objectives

The purpose of this assignment is to help you:

- Refresh knowledge on string, list and dictionary.
- Do a little bit study on list comprehension and dictionary comprehension.
- Learn to use zip function in Python.

## Background

### One simple substitution cipher

In this project, we will write some code to encode and decode messages. There is a simple kind of coding scheme called substitution cipher in which every letter of the alphabet is mapped to a different letter. A simple case of this might be described as follows:

```
alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ  
codestring: JKLMNWXCDPQRSTUVAEFOBGHIZY
```

Each letter in the top line (aka. in the alphabet) will get changed into the corresponding letter in the bottom line (aka. in the codestring). For example, the string `HELLO` is encoded as `CNRRU` using the code above. The string `GDL0UEZ` is decoded as `VICT0RY`. We are calling the string to be encoded (e.g. `HELLO`) and the decoded result (e.g. `VICT0RY`) plaintext. And we are calling the encoded string (e.g. `CNRRU`) and the string to be decoded (e.g. `GDL0UEZ`) ciphertext.

The alphabet rarely changes and people usually change codestring for different cipher applications. In this assignment, if not specified, the alphabet will always be solid as `ABCDEFGHIJKLMNOPQRSTUVWXYZ` and the alphabet and codestring are with the same length.

# Assignment

When you are reading this assignment, you must have already downloaded the skeleton zip file. In the zip file, you can find a skeleton code, named cipher.py. All you need to do is to complete the skeleton code based on the instructions and submit it to the Mimir system.

## Substitution cipher with string operation

Before we are going to work on the skeleton code, we will first try some simple implementations of substitution cipher. It is not too hard to decode such a code if you know the ciphertext for the whole alphabet. Here is a little code that prints a decoded output.

```
alphabet = "ABCDEFGHJKLMNOPQRSTUVWXYZ"
codestring = "BCDEFGHIJKLMNOPQRSTUVWXYZA"
ciphertext = "IFMPXPSME"
for char in ciphertext:
    print(alphabet[codestring.index(char)], end = "")
```

### Output

```
HELLOWORLD
```

If you wanted to produce a plaintext but not print the result, you might try something like the following:

```
plaintext = ""
for char in ciphertext:
    plaintext = plaintext + alphabet[codestring.index(char)]
```

Now that you know how the decode function for string operation works, write the encode function.

```
encode_string(codestring, plaintext)
decode_string(codestring, ciphertext)
```

Note: You must complete both the encode and the decode functions.

## Some other concerns

The first thing is to adapt our code so that it automatically converts plaintext, ciphertext, and codestring to uppercase, where we can use the `str.upper()` method. This method returns an uppercase version of the string. For example, `'Hello'.upper()` returns `HELLO`.

The second thing we need to consider is that the encode and decode functions should leave all punctuation marks in place. This will mean that you should check if the letter is in the code or its inverse and leave the letter if not. Checking if a key `k` is in a dictionary `D` can be done by writing `if k in D`. Moreover, if there are spaces in the plaintext, we should change the spaces into dashes (-) in the ciphertext, and vice versa.

After implementing the two things above, if we have a certain codestring

`JMBCYEKLFDGUVWHINXRTOSPZQA`, given plaintext `Ab3c, De1::6`, the encoded ciphertext should be `JM3B,-CY1::6`. With the same codestring, if ciphertext `A-p4s#%!` is given, the decoded plaintext should be `Z W4V#%!`.

## Submit your work to Mimir

Submit your code to Mimir after you complete your code. Mimir will automatically grade your submission based on different unit test cases (with different codestring, plaintext and ciphertext). You can submit your code to Mimir any number of times to refresh your existing score before the submission deadline.