Classical Ciphers

Caeser Cipher:

This Cipher type depends on shitting the letters of the message by value of key in mod 26

```
def CaeserCipher(caeserPlaintext, caeserKeys)
```

This function take two parameters: 1) `caeserPlaintext` :our message
2) `caeserKeys`:the key value we will shit the letters of our messages with.

This function will return cipher text in list and we will use `def convert(s)`
To return string

PlayFair Cipher

This Cipher type depends on generating matrix from letters which are letters of key and the rest of alphabet, then take two letters and get the intersection letter in the matrix between the column and row vice versa.

```
def generatematrix(key)
```

This function take the key and generate the matrix.

```
def MapDiagraph(Plaintext)
```

This function take plaintext and splits it to two letters groups.

```
def Letter_coordinates(key_matrix, character)
```

This function get the position of each letter.

```
def encrypt(message, key)
```

This function make the playfair encryption using the above three functions.

Hill Cipher:

This type of cipher depends on convert message to ascii code matrix and take a matrix of numbers as a key either 2x2 or 3x3 and make encryption.

```
def Encryption(message, key)
```

This function take the plaintext and the key and make the whole encryption and return the cipher text.

Vigenere Cipher:

This Cipher type depends on generating key either auto (true mode, concatenate the key letters with some letters of message till the both key and message length become equal) or repeating (false mode, repeating key letters till the both key and message length become equal), and XOR the bits of generated key and plaintext to get the cipher text.

```
def GetKey(string, key, Mood)
```

This function take plaintext, the key, and the mode of generating key , and return the suitable key.

```
def vigenere_cipher(string, key)
```

This function take plaintext and the key to return the cipher text.

Vernam Cipher:

This Cipher type depends on generating a random key and XOR it's letters with the message letters.

```
def VernamCipherFunction(text, key)
```

This function take the plaintext and a random key from user and make the whole encryption and return the cipher text.