DATE: 25-08-2022

LAB 01: Working with classical ciphers

NAME	Vishwa Mehul Mehta
SRN	PES2UG20CS389
SECTION	F

For the given questions, write a python code and attach the snapshots.

1.	For the given input, perform Caesar cipher encryption and decryption. Plain text: “CRYPTOGRAPHY” Key: 10

SOL	**Code:**
	
	ALPHABET = "abcdefghijklmnopqrstuvwxyz"
	
	def encode(text: str, n: int) -> str:
	  x = n % 26
	  rotalpha = ALPHABET[-x:] + ALPHABET[:-x]
	  decoded = ""
	  for alpha in text:
	  if alpha == " ":
	  decoded += " "
	  else:
	  decoded += ALPHABET[rotalpha.index(alpha)]
	  return decoded
	
	def decode(text: str, n: int) -> str:
	  x = n % 26
	  rotalpha = ALPHABET[-x:] + ALPHABET[:-x]
	  encoded = ""
	  for alpha in text:
	  if alpha == " ":
	  encoded += " "
	  else:
	  encoded += rotalpha[ALPHABET.index(alpha)]
	  return encoded
	
	text = input("Enter text: ")
	n = int(input("Enter the value for rotate: "))
	encrypted = encode(text.lower(), n)
	print("Encoded Text: " + encrypted.upper())
	decrypted = decode(encrypted.lower(), n)
	print("Decoded Text: " + decrypted.upper())

	Screenshot:
	
	`vishwa@Acer-Vishwa:/mnt/c/Users/Vishwa/Documents/Vishwa_PES/Sem5/314_AC/week1$ python3 caesar_cipher.py`
	`Enter text: cryptography`
	`Enter the value for rotate: 10`
	`Encoded Text: MBIZDYQBKZRI`
	`Decoded Text: CRYPTOGRAPHY`
	`vishwa@Acer-Vishwa:/mnt/c/Users/Vishwa/Documents/Vishwa_PES/Sem5/314_AC/week1$`

2.	For the plaintext given in question 1, apply Play Fair cipher encryption with key “WORK”.

SOL	**Code:**
	
	text = input("Enter text: ")
	key = input("Enter the key: ")
	
	ALPHABET = "abcdefghiklmnopqrstuvwxyz"
	KEY = key
	#KEY = "work"
	NEW_ALPHA = list(ALPHABET)
	for i in KEY:
	  if i in NEW_ALPHA:
	  NEW_ALPHA.remove(i)
	  else:
	  continue
	NEW_ALPHA = ''.join(NEW_ALPHA)
	NEW_ALPHA = KEY + NEW_ALPHA
	#print(NEW_ALPHA)
	MATRIX = []
	
	for i in range(0,25,5):
	  MATRIX.append(list(NEW_ALPHA[i:i+5]))
	#print(MATRIX)
	
	#text = "cryptographm"
	def make_digraph(text):
	  di = []
	  text = text.replace('j', 'i')
	  for i in range(0, len(text), 2):
	  di.append(list(text[i:i+2]))
	  if len(text) % 2 == 1:
	  di[-1].append("j")
	  #print(di)
	  return di
	
	def find_index(pair):
	  for i in range(5):
	  for j in range(5):

```
            if MATRIX[i][j] == pair[0]:
                i1 = [i, j]
            if MATRIX[i][j] == pair[1]:
                i2 = [i, j]
    return [i1, i2]

def encode(text):
    di = make_digraph(text)
    encoded_text = []
    for i in di:
        i1, i2 = find_index(i)
        if abs(i1[0] - i2[0]) != 0 and abs(i1[1] - i2[1]) != 0:
            if abs(i1[0] - i2[0]) <= abs(i1[1] - i2[1]):
                new_i1 = [i1[0], i2[1]]
                new_i2 = [i2[0], i1[1]]
            elif abs(i1[0] - i2[0]) >= abs(i1[1] - i2[1]):
                new_i1 = [i1[1], i2[0]]
                new_i2 = [i2[1], i1[0]]
        elif abs(i1[0] - i2[0]) == 0:
            if i1[1]+1 > 4:
                i1[1] -= 4
            else:
                i1[1] += 1
            if i2[1]+1 > 4:
                i2[1] -= 4
            else:
                i2[1] += 1
            new_i1 = [i1[0], i1[1]]
            new_i2 = [i2[0], i2[1]]
        elif abs(i1[1] - i2[1]) == 0:
            if i1[0]+1 > 4:
                i1[0] -= 4
            else:
                i1[0] += 1
            if i2[0]+1 > 4:
                i2[0] -= 4
            else:
                i2[0] += 1
            new_i1 = [i1[0], i1[1]]
            new_i2 = [i2[0], i2[1]]

        encoded_text.append(MATRIX[new_i1[0]][new_i1[1]])
        encoded_text.append(MATRIX[new_i2[0]][new_i2[1]])
    #print(''.join(encoded_text))
    return ''.join(encoded_text).upper()

def decode(text):
```

```
    di = make_digraph(text)
    decoded_text = []
    for i in di:
        i1, i2 = find_index(i)
        if abs(i1[0] - i2[0]) != 0 and abs(i1[1] - i2[1]) != 0:
            if abs(i1[0] - i2[0]) <= abs(i1[1] - i2[1]):
                new_i1 = [i1[0], i2[1]]
                new_i2 = [i2[0], i1[1]]
            elif abs(i1[0] - i2[0]) >= abs(i1[1] - i2[1]):
                new_i1 = [i1[1], i2[0]]
                new_i2 = [i2[1], i1[0]]
        elif abs(i1[0] - i2[0]) == 0:
            new_i1 = [i1[0], i1[1]-1]
            new_i2 = [i2[0], i2[1]-1]
        elif abs(i1[1] - i2[1]) == 0:
            new_i1 = [i1[0]-1, i1[1]]
            new_i2 = [i2[0]-1, i2[1]]

        decoded_text.append(MATRIX[new_i1[0]][new_i1[1]])
        decoded_text.append(MATRIX[new_i2[0]][new_i2[1]])
    #print(''.join(decoded_text))
    return ''.join(decoded_text).upper()

encoded = encode(text.lower())
print("Encrypted Text: " + encoded)
decoded = decode(encoded.lower())
print("Decrypted Text: " + decoded)
```

Screenshot: