**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:** |
| 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:** |