DATE: 20-08-2025

LAB 01: Working with classical ciphers

NAME	Keerthan pv
SRN	PES2UG23CS272
SECTION	E

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:**
	def caesarencrypt(plaintext,key):
	  result=" "
	  for ch in plaintext:
	  if ch.isalpha():
	  result+=chr((ord(ch)-65+key)%26+65)
	  else:
	  result+=ch
	  return result
	
	def caesardecrypt(ciphertext,key):
	  result=" "
	  for ch in ciphertext:
	  if ch.isalpha():
	  result+=chr((ord(ch)-65-key)%26+65)
	  else:
	  result+=ch
	  return result
	
	plaintext="CRYPTOGRAPHY"
	key=10
	cipher=caesarencrypt(plaintext,key)
	decrypted=caesardecrypt(cipher,key)
	print("Plaintext:",plaintext)
	print("Ciphertext:", cipher)
	print("Decrypted:",decrypted)
	
	Screenshot:

```
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-5\AC LAB> python lab-one.py
Plaintext: CRYPTOGRAPHY
Ciphertext:  MBIZDYQBKZRI
Decrypted:   CRYPTOGRAPHY
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-5\AC LAB>
```

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

SOL

Code:

```
def playfair_matrix(key):
    key=key.upper().replace("J","I")
    matrix,used=[],set()
    for ch in key:
        if ch.isalpha() and ch not in used:
            matrix.append(ch); used.add(ch)
    for ch in "ABCDEFGHIKLMNOPQRSTUVWXYZ":
        if ch not in used:
            matrix.append(ch); used.add(ch)
    return [matrix[i: i+5] for i in range(0,25,5)]

def findposition(matrix,ch):
    for i, row in enumerate(matrix):
        for j, val in enumerate(row):
            if val==ch:return i,j

def playfair_prepare(text):
    text=text.upper().replace("J","I")
    i,result=0,""
    while i<len(text):
        a=text[i]
        b=text[i+1] if i+1<len(text) else "X"
        if a==b:
            result+=a+"X";i+=1
        else:
            result+=a+b;i+=2
    return result

def playfair_encrypt(plaintext,key):
    matrix= playfair_matrix(key)
    prepared=playfair_prepare(plaintext)
    cipher=""
    for i in range(0,len(prepared),2):
        a,b=prepared[i],prepared[i+1]
        r1,c1=findposition(matrix,a)
        r2,c2=findposition(matrix,b)
        if r1==r2:
            cipher+=matrix[r1][(c1+1)%5]+matrix[r2][(c2+1)%5]
        elif c1==c2:
```

```
            cipher+=matrix[(r1+1)%5][c1]+matrix[(r2+1)%5][c2]
        else:
            cipher+=matrix[r1][c2]+matrix[r2][c1]
    return cipher

plaintext="CRYPTOGRAPHY"
key="WORK"
cipher=playfair_encrypt(plaintext,key)
print("Plaintext:",plaintext)
print("Ciphertext:",cipher)
```

Screenshot:

```
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-5\AC LAB> python lab-one-two.py
Plaintext: CRYPTOGRAPHY
Ciphertext: DOVSPAIWOTLV
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-5\AC LAB>
```

3. For the plaintext= “WORK”, apply Hill cipher cipher encryption with key = [1,2; 2;2].

SOL

Code:

```
import numpy as np
def hillencrypt(plaintext,keymatrix):
    n=len(keymatrix)
    text=plaintext.upper().replace(" ","")
    while len(text)% n!=0:
        text+="X"
    cipher=" "
    for i in range(0,len(text),n):
        block=np.array([ord(ch)-65 for ch in text[i: i+n]])
        res=np.dot(keymatrix,block)%26
        cipher+="".join(chr(r+65) for r in res)
    return cipher

plaintext="WORK"
key_matrix=np.array([[1,2],[2,2]])
cipher=hillencrypt(plaintext,key_matrix)

print("Plaintext -",plaintext)
print("Ciphertext -",cipher)
```

Screenshot:

```
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-5\AC LAB> python lab-one-three.py
Plaintext - WORK
Ciphertext -  YULC
PS C:\Users\keert\OneDrive\Desktop\CSE\SEM-5\AC LAB>
```