Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Experiment 1

Date of Performance : 25 February 2023 **Date of Submission:** 26 February 2023

SAP Id: 60004200132 **Name :** Ayush Jain

Div: B **Batch :** B3

Aim of Experiment

Design and Implement Encryption and Decryption Algorithm for Caesar cipher cryptographic algorithm by considering letter [A..Z] and digits [0..9]. Create two functions Encrypt() and Decrypt(). Apply Brute Force Attack to reveal secret. Create Function BruteForce().

(CO1)

Theory / Algorithm / Conceptual Description

The Caesar Cipher technique is one of the earliest and simplest methods of encryption technique. It's simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter with a fixed number of positions down the alphabet. For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.

Thus to cipher a given text we need an integer value, known as a shift which indicates the number of positions each letter of the text has been moved down. The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,..., Z = 25. Encryption of a letter by a shift n can be described mathematically as.

$$E_n(x) = (x + n) mod\ 26$$

(Encryption Phase with shift n)

$$D_n(x) = (x - n) mod\ 26$$

(Decryption Phase with shift n)

The same logic can be applied to numeric values (0…9) in the text.

Program

```
#encoding
def Encrypt():
     print("****Encoding****")
     PT = input("Print Plain text : ")
     key = int(input("Print Key : "))
     CT = ""

     for ch in PT:
         if ord(ch)>=48 and ord(ch)<=57:
            temp = chr(ord(ch)+key%10)
         else:
            temp = chr(ord(ch)+(key%26))
         CT=CT+temp
     print('Ciphered text : ',CT)

 #decoding
def Decrypt():
     print("\n****Decoding****")
     CT = input("Print Ciphered text : ")
     key = int(input("Print Key : "))
     PT = ""

     for ch in CT:
         if ord(ch)>=48 and ord(ch)<=57:
            temp = chr(ord(ch)-(key%10))
         else:
            temp = chr(ord(ch)-(key%26))
         PT=PT+temp
     print('Plain text : ',PT)


#brute force encoding
def BruteForce():
     print("\n****Brute force approach****")
     PT = input("Print Plain text : ")

     print("Key"+" "+"Ciphered Text")
     for key in range(0,26):
        CT = ""
        for ch in PT:
            if ord(ch)>=48 and ord(ch)<=57:
```

```
                temp = (ord(ch)+(key%10))
                if temp not in range(48,58):
                    temp = temp-10
                temp = chr(temp)
            else:
                temp = (ord(ch)+(key%26))
                if temp not in range(65,91) and temp not in range(97,123):
                        temp = temp - 26
                temp = chr(temp)
            CT=CT+temp
        print(str(key)+" "+CT)

Encrypt()
Decrypt()
BruteForce()
```

Input

1. A String of Alphabet and Digits.
2. An Integer between 0-25 denoting the required shift.

Encrypt()
PT = ABC012
Key = 3

Decrypt()
CT = DEF345
Key = 3

BruteForce()
PT = ABC012

Output

```
****Encoding****
Print Plain text : ABC123
Print Key : 3
Ciphered text :  DEF456

****Decoding****
Print Ciphered text : DEF456
Print Key : 3
Plain text :  ABC123

****Brute force approach****
Print Plain text : ABC123
Key Ciphered Text
0 ABC123
1 BCD234
2 CDE345
3 DEF456
4 EFG567
5 FGH678
6 GHI789
7 HIJ890
8 IJK901
9 JKL012
10 KLM123
11 LMN234
12 MNO345
13 NOP456
14 OPQ567
15 PQR678
16 QRS789
17 RST890
18 STU901
19 TUV012
20 UVW123
21 VWX234
22 WXY345
23 XYZ456
24 YZA567
25 ZAB678
```