Rashtriya Raksha University

School of Information Technology, Artificial Intelligence & Cyber Security (SITAICS)

At- Lavad, Dahegam, Gandhinagar, Gujarat-382305

Practical File

(Introduction to Cryptography)

Name: Sarthak Sanay
Enrollment No: 230031101611051
Subject Name: Introduction to Cryptography
Subject Code: G4A19ITC
Program: B.Tech CSE (with specialization in Cyber Security)
Year: 2nd year (Semester-IV)

This is certifying that Mr. Sarthak Sanay has satisfactorily completed all experiments in the practical work prescribed by SITAICS in the ITC laboratory.

Dr. Ashish Revar
SUBJECT INCHARGE

PRACTICAL - 1

AIM: TO IMPLEMENT CAESAR CIPHER

BRIEF :-

The Caesar cipher, named after the Roman general Julius Caesar, is one of the oldest and simplest encryption techniques ever devised.

At its core, the cipher replaces each letter in a message (the plaintext) with the letter a fixed number of positions down the alphabet. This fixed number is known as the "key." For example, with a key of 3, A becomes D, B becomes E, C becomes F, and so on; after Z it wraps around back to A. To encrypt, you add the key to each letter's position; to decrypt, you subtract the key, using modulo 26 arithmetic to handle the wrap-around.

Non-letter characters - such as spaces, digits, and punctuation—are typically left unchanged, which makes the cipher easy to implement in code or by hand. Because the only secret is the key (an integer between 1 and 25), there are only 25 possible non-trivial shifts. An attacker can therefore mount a brute-force attack—trying all possible keys - or apply simple frequency analysis to recover the original message.

Despite its historical importance and pedagogical value in introducing concepts like modular arithmetic and substitution ciphers, the Caesar cipher offers no real security by modern standards. Its ease of breaking makes it unsuitable for protecting sensitive data today, but it remains a popular example in cryptography tutorials and puzzles.

ALGORITHM / PSEUDOCODE :-

```
repeat
  print menu
  read ch
  if ch == 1 then
    read plain_text
    read key
    cipher_text = ""
    for each c in plain_text do
      if isUpper(c) then
        cipher_text += ( (c-'A'+key) mod 26 ) + 'A'
      else if isLower(c) then
        cipher_text += ( (c-'a'+key) mod 26 ) + 'a'
      else
        cipher_text += c
    end for
    print cipher_text

  else if ch == 2 then
    read cipher_text
    read key
    plain_text = ""
    for each c in cipher_text do
      if isUpper(c) then
        plain_text += ( (c-'A'-key+26) mod 26 ) + 'A'
      else if isLower(c) then
        plain_text += ( (c-'a'-key+26) mod 26 ) + 'a'
      else
        plain_text += c
    end for
    print plain_text

  else if ch == 0 then
    exit loop
  else
    print "Invalid choice"
  end if
until ch == 0
```

CODE :-

```
print("\nCaesar Cipher Encryption & Decryption Tool:-")
ch = 1

while (ch!=0):
    ch = int(input("\nEnter 1 to Encrypt. \nEnter 2 to Decrypt.
\nEnter 0 to Exit. \nEnter choice: "))

    match ch:

        case 1:
            print("\nEncrypting Caesar Cipher!\n")
            plain_text = str(input("Enter plain text: "))
            key = int(input("Enter key: "))
            cipher_text = ""

            for i in range(0, len(plain_text)):
                char = plain_text[i]

                if char == chr(32):
                    cipher_text += char
                    continue

                elif (char.isupper()):
                    cipher_text += chr((ord(char) + key-65) % 26 + 65)

                elif (char.islower()):
                    cipher_text += chr((ord(char) + key-97) % 26 + 97)

                else:
                    cipher_text += char

            print("Plain Text:  ", plain_text)
            print("Cipher Text: ", cipher_text, "\n")


        case 2:
            print("\nDecrypting Caesar Cipher!\n")
            cipher_text = str(input("Enter cipher text: "))
            key = int(input("Enter key: "))
            plain_text = ""

            for i in range(0, len(cipher_text)):
                char = cipher_text[i]

                if char == chr(32):
                    plain_text += char
                    continue
```

```
            elif (char.isupper()):
                plain_text += chr((ord(char) - key-65) % 26 + 65)

            elif (char.islower()):
                plain_text += chr((ord(char) - key-97) % 26 + 97)

            else:
                plain_text += char

        print("Cipher Text: ", cipher_text)
        print("Plain Text:  ", plain_text, "\n")


    case 0:
        print("\nProgram exited successfully!")

    case _:
        print("\nEnter correct choice!\n")
```

OUTPUT :-

```
Caesar Cipher Encryption & Decryption Tool:

Enter 1 to Encrypt.
Enter 2 to Decrypt.
Enter 0 to Exit.
Enter choice: 1

Encrypting Caesar Cipher!

Enter plain text: Hello, My name is Sarthak
Enter key: 17
Plain Text:  Hello, My name is Sarthak
Cipher Text: Yvccf, Dp erdv zj Jrikyrb

Enter 1 to Encrypt.
Enter 2 to Decrypt.
Enter 0 to Exit.
Enter choice: 2

Decrypting Caesar Cipher!

Enter cipher text: Yvccf, Dp erdv zj Jrikyrb
Enter key: 17
Cipher Text:  Yvccf, Dp erdv zj Jrikyrb
Plain Text:   Hello, My name is Sarthak

Enter 1 to Encrypt.
Enter 2 to Decrypt.
Enter 0 to Exit.
Enter choice: 0

Program exited successfully!
```