21CSE1003 | ASHISH SINGH

Q1. Write a program to encrypt the message "Are you Ready for class" using shift cipher with any key value. Then decrypt the message back to plain text.

Algorithm:

Shift Cipher Encryption (shift_cipher)

1. **Input:**
    - A string plainText that represents the message to be encrypted.
    - An integer key representing the shift value for the Caesar cipher.
2. **Initialize:**
    - Create an empty list c to hold the encrypted characters.
3. **For each character i in the plainText:**
    - **If i is an alphabetic character (i.isalpha()):**
        - **If i is lowercase (i.islower()):**
            1. Convert the character i to its alphabetical index relative to 'a' by subtracting ord('a').
            2. Add the shift key to the index.
            3. Perform modulo 26 to ensure the index wraps around within the range of lowercase letters (0-25).
            4. Convert the result back to a character by adding ord('a').
            5. Append the shifted character to the list c.
        - **If i is uppercase (i.isupper()):**
            1. Convert the character i to its alphabetical index relative to 'A' by subtracting ord('A').
            2. Add the shift key to the index.
            3. Perform modulo 26 to ensure the index wraps around within the range of uppercase letters (0-25).
            4. Convert the result back to a character by adding ord('A').
            5. Append the shifted character to the list c.
    - **If i is not alphabetic (e.g., spaces, punctuation, etc.):**
        - Append the character unchanged to the list c.
4. **Join all the elements of list c** into a single string.
5. **Return** the final encrypted string (cipherText).

Shift Cipher Decryption (shift_dechiper)

1. **Input:**
    - A string cipherText that represents the message to be decrypted.
    - An integer key representing the shift value used during encryption.
2. **Initialize:**
    - Create an empty list p to hold the decrypted characters.
3. **For each character i in the cipherText:**
    - **If i is an alphabetic character (i.isalpha()):**
        - **If i is lowercase (i.islower()):**
            1. Convert the character i to its alphabetical index relative to 'a' by subtracting ord('a').
            2. Subtract the shift key from the index.
            3. Perform modulo 26 to ensure the index wraps around within the range of lowercase letters (0-25).
            4. Convert the result back to a character by adding ord('a').
            5. Append the shifted character to the list p.
        - **If i is uppercase (i.isupper()):**
            1. Convert the character i to its alphabetical index relative to 'A' by subtracting ord('A').
            2. Subtract the shift key from the index.
            3. Perform modulo 26 to ensure the index wraps around within the range of uppercase letters (0-25).
            4. Convert the result back to a character by adding ord('A').
            5. Append the shifted character to the list p.
    - **If i is not alphabetic (e.g., spaces, punctuation, etc.):**
        - Append the character unchanged to the list p.
4. **Join all the elements of list p** into a single string.
5. **Return** the final decrypted string (plainText).

LAB_5\shift_cipher.py

```python
# Ashish Singh
# 21CSE1003

# Q1. Write a program to encrypt the message "Are you Ready for class" using shift cipher
# with any key value. Then decrypt the message back to plain text.

def shift_cipher(plainText, key):
    c = []
    for i in plainText:
        if i.isalpha():  # Only shift alphabetic characters
            # For lowercase letters
            if i.islower():
                shifted_char = chr((ord(i) - ord('a') + key) % 26 + ord('a'))
            # For uppercase letters
            elif i.isupper():
                shifted_char = chr((ord(i) - ord('A') + key) % 26 + ord('A'))
            c.append(shifted_char)
        else:
            # Non-alphabet characters are not shifted
            c.append(i)
    return ''.join(c)

def shift_dechiper(chipherText, key):
    p = []
    for i in chipherText:
        if i.isalpha():  # Only shift alphabetic characters
            # For lowercase letters
            if i.islower():
                shifted_char = chr((ord(i) - ord('a') - key) % 26 + ord('a'))
            # For uppercase letters
            elif i.isupper():
                shifted_char = chr((ord(i) - ord('A') - key) % 26 + ord('A'))
            p.append(shifted_char)
        else:
            # Non-alphabet characters are not shifted
            p.append(i)
    return ''.join(p)

# plainText = input("\nEnter plaintext: ")
plainText = "Are you Ready for class"
key = int(input("\nEnter key: "))

cipherText = shift_cipher(plainText, key)
plainText = shift_dechiper(cipherText, key)

print(f"\nCipherText: {cipherText}")
print(f"PlainText: {plainText}\n")
```

Q2. Write a program to find the key value of the given cipher text (JBCRCLQRWCRVNBJENBWRWN)

Algorithm:

Decryption (shift_dechiper)

1. **Input:**
    - A string cipherText representing the encrypted text.
    - An integer key representing the Caesar cipher shift value.
2. **Initialize:**
    - Create an empty list p to hold the decrypted characters.
3. **For each character i in the cipherText:**
    - **If i is an alphabetic character (i.isalpha()):**
        - **If i is lowercase (i.islower()):**
            1. Convert the character i to its alphabetical index relative to 'a' by subtracting ord('a').
            2. Subtract the shift key from the index.
            3. Perform modulo 26 to ensure the index wraps around within the range of lowercase letters (0-25).
            4. Convert the result back to a character by adding ord('a').
            5. Append the shifted character to the list p.
        - **If i is uppercase (i.isupper()):**
            1. Convert the character i to its alphabetical index relative to 'A' by subtracting ord('A').
            2. Subtract the shift key from the index.
            3. Perform modulo 26 to ensure the index wraps around within the range of uppercase letters (0-25).
            4. Convert the result back to a character by adding ord('A').
            5. Append the shifted character to the list p.
    - **If i is not alphabetic (e.g., spaces, punctuation, etc.):**
        - Append the character unchanged to the list p.
4. **Join all the elements of list p** into a single string.
5. **Return** the decrypted string (plainText).

Brute-Force Decryption Loop

1. **Input:**
    - A string cipherText representing the encrypted message.
    - A list of potential keys ranging from 0 to 25 (inclusive).
2. **For each key k in the range of 0 to 25:**
    - Call the shift_dechiper function with the cipherText and the current key k.
    - Print the value of k.
    - Print the resulting decrypted plainText for that key.

LAB_5\find_the_key.py

```python

# Q2. Write a program to find the key value of the given cipher text:
# (JBCRCLQRWCRVNBJENBWRWN)

def shift_dechiper(chipherText, key):
    p = []
    for i in chipherText:
        if i.isalpha():  # Only shift alphabetic characters
            # For lowercase letters
            if i.islower():
                shifted_char = chr((ord(i) - ord('a') - key) % 26 + ord('a'))
            # For uppercase letters
            elif i.isupper():
                shifted_char = chr((ord(i) - ord('A') - key) % 26 + ord('A'))
            p.append(shifted_char)
        else:
            # Non-alphabet characters are not shifted
            p.append(i)
    return ''.join(p)

# plainText = input("\nEnter plaintext: ")
cipherText = "JBCRCLQRWCRVNBJENBWRWN"
key = [i for i in range(0, 26)]

for k in key:
    print(f"For key = {k}")
    plainText = shift_dechiper(cipherText, k)

    print(f"PlainText: {plainText}\n")
```

Q3. Substitution cipher algorithm.

Algorithm:

1. **Define Substitution Function:**
    - Input: plainText (text to be encrypted), keyDict (dictionary mapping letters to substitutions).
    - For each character i in plainText, append its corresponding substitution from keyDict to a list c.
    - Join the list c to form the final encrypted string and return it.
2. **Initialize the Dictionary d:**
    - Create a dictionary d where each uppercase letter (A-Z) is mapped to an underscore ('_'), indicating the substitution is not yet assigned.
3. **For each key in d:**
    - Ask the user to input a substitution for the current key.
    - If the substitution has already been assigned in d.values(), print a message indicating it is already in use and ask for a new one.
    - Otherwise, store the new substitution in the dictionary d and convert it to uppercase.
4. **Print the final substitution dictionary.**
5. **Encrypt the Plain Text:**
    - Prompt the user to enter the plainText and convert it to uppercase.
    - Call the substitution function with plainText and the dictionary d to get the encrypted cipherText.
6. **Output the Encrypted Text.**

LAB_5\substitution.py

```python

# Q3. Substitution cipher implementation
def substitution(plainText, keyDict):
    c = []
    for i in plainText:
        c.append(keyDict[i])
    return ''.join(c)

d = {'A': '_',
     'B': '_',
     'C': '_',
     'D': '_',
     'E': '_',
     'F': '_',
     'G': '_',
     'H': '_',
     'I': '_',
     'J': '_',
     'K': '_',
     'L': '_',
     'M': '_',
     'N': '_',
     'O': '_',
     'P': '_',
     'Q': '_',
     'R': '_',
     'S': '_',
     'T': '_',
     'U': '_',
     'V': '_',
     'W': '_',
     'X': '_',
     'Y': '_',
     'Z': '_'}

for _ in d.keys():
    t = input(f"Enter substitution for {_}: ")
    if t in d.values():
        print("substitution already assigned")
    else:
        d[_] = t.upper()

print(f"\n{d}\n")

plainText = input("\nEnter plain text: ").upper()
cipherText = substitution(plainText, d)

print(f"Encrypted text: {cipherText}\n")
```

Q4. Write shift cipher considering numbers also.

Algorithm:

Initialization:

1. **Z36 List:**
    - A list Z36 contains characters '0'-'9' and 'A'-'Z'. This represents the 36 symbols (digits and uppercase letters) that can be shifted in the cipher.

Shift Cipher Function (shift_cipher):

1. **Input:**
    - plainText: The message to be encrypted (e.g., "i transfered rs 2034 to you").
    - key: The integer value by which characters are shifted.
2. **Process:**
    - Initialize an empty list c to store the encrypted characters.
    - For each character i in the plainText:
        - **If i is a letter:**
            - **For lowercase letters:** Convert i to uppercase, find its position in Z36, shift it by the key, and append the corresponding character from Z36 to c.
            - **For uppercase letters:** Find i's position in Z36, shift it by the key, and append the corresponding character to c.
        - **If i is a digit:** Find its position in Z36, shift it by the key, and append the shifted character to c.
        - **If i is a non-alphabetical/non-digit character:** Append i unchanged to c (e.g., spaces, punctuation).
3. **Output:**
    - Join the list c into a single string (the encrypted text) and return it.

For decryption just subtract the key from the Z36 value.

LAB_5\shift_cipher_2 copy.py

```python
# Q1. Write a program to encrypt the message "i transfered rs 2034 to you" using shift cipher
# with any key value. Then decrypt the message back to plain text.

Z36 = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']

def shift_cipher(plainText, key):
    c = []
    for i in plainText:
        if i.isalpha():  # Only shift alphabetic characters
            if i.islower():
                shifted_char = Z36[(Z36.index(i.upper()) + key) % 36]
            elif i.isupper():
                shifted_char = Z36[(Z36.index(i) + key) % 36]
            c.append(shifted_char)
        elif i.isdigit():
                shifted_char = Z36[(Z36.index(i) + key) % 36]
                c.append(shifted_char)
        else:
            c.append(i)
    return ''.join(c)

def shift_dechiper(chipherText, key):
    p = []
    for i in chipherText:
        if i.isalpha():  # Only shift alphabetic characters
            if i.islower():
                shifted_char = Z36[(Z36.index(i.upper()) - key) % 36]
            elif i.isupper():
                shifted_char = Z36[(Z36.index(i) - key) % 36]
            p.append(shifted_char)
        elif i.isdigit():
                shifted_char = Z36[(Z36.index(i) - key) % 36]
                p.append(shifted_char)
        else:
            p.append(i)
    return ''.join(p)

plainText = "i transfered rs 2034 to you"
key = int(input("\nEnter key: "))

cipherText = shift_cipher(plainText, key)
plainText_2 = shift_dechiper(cipherText, key)

print(f"\nCipherText: {cipherText}")
print(f"PlainText: {plainText_2}\n")
```