**Assignment-1**

**Pranav Murali**

**A20555824**

**ITIS Assignment 1.py file is also attached**

**Program:-**

# Intro to Infosec

# Assignment 1

# Pranav Murali

# A20555824

# Encryption function that performs a shift cipher on the given plaintext

def encrypt(plaintext, key):

    if not plaintext:

        return "No character found. Please enter valiid plaintext."

    ciphertext = ""

    # Loop through each character in the plaintext

    for char in plaintext:

        # Check if the character is an alphabet letter

        if char.isalpha():

            # Determine the shift based on uppercase or lowercase

            shift = 65 if char.isupper() else 97

            # Apply the shift and append the encrypted character to the ciphertext

            encrypted\_char = chr((ord(char) - shift + key) % 26 + shift)

            ciphertext += encrypted\_char

        else:

            # If the character is not an alphabet letter, keep it unchanged1

            ciphertext += char

    # Return the resulting ciphertext

    return ciphertext

# Decryption function that reverses the shift cipher to retrieve the original plaintext

def decrypt(ciphertext, key):

    if not ciphertext:

        return "No character found. Please enter valid ciphertext."

    decrypted\_text = ""

    # Loop through each character in the ciphertext

    for char in ciphertext:

        # Check if the character is an alphabet letter1

        if char.isalpha():

            # Determine the shift based on uppercase or lowercase

            shift = 65 if char.isupper() else 97

            # Reverse the shift and append the decrypted character to the result

            decrypted\_char = chr((ord(char) - shift - key) % 26 + shift)

            decrypted\_text += decrypted\_char

        else:

            # If the character is not an alphabet letter, keep it unchanged

            decrypted\_text += char

    # Return the resulting decrypted plaintext

    return decrypted\_text

# Brute force attack function that tries all possible keys for a Caesar cipher

def brute\_force\_attack(ciphertext):

    if not ciphertext:

        return "No character found. Please enter valid ciphertext."

    print("Brute Force Attack Results:")

    # Loop through all possible keys

    for possible\_key in range(1, 26):

        # Decrypt the ciphertext using the current key

        decrypted\_text = decrypt(ciphertext, possible\_key)

        # Print the result for the current key

        print(f"Key {possible\_key}: {decrypted\_text}")

# Main program loop

while True:

    # Display the program features menu

    print("Program Features:")

    print("1. Encryption")

    print("2. Decryption")

    print("3. Brute Force Attack")

    print("4. Exit")

    # Get user input for the chosen option

    choice = input("Enter the option number: ")

    # Check the user's choice and execute the corresponding functionality

    if choice == "1":

        plaintext = input("Enter the plaintext: ")

        # Validate if the key is a valid integer

        try:

            key = int(input("Enter the key for encryption: "))

        except ValueError:

            print("Invalid input. Key must be a numeric value.")

            continue

        ciphertext = encrypt(plaintext, key)

        print("Encrypted message:", ciphertext)

    elif choice == "2":

        ciphertext = input("Enter the ciphertext: ")

        # Validate if the key is a valid integer

        try:

            key = int(input("Enter the key for decryption: "))

        except ValueError:

            print("Invalid input. Key must be a numeric value.")

            continue

        decrypted\_text = decrypt(ciphertext, key)

        print("Decrypted message:", decrypted\_text)

    elif choice == "3":

        ciphertext = input("Enter the ciphertext for brute force attack: ")

        brute\_force\_attack(ciphertext)

    elif choice == "4":

        print("Exiting the program.")

        break

    else:

        print("Invalid option. Please choose a valid option.")

**Output:-** A screenshot of a computer

Description automatically generated

A black background with a black square

Description automatically generated with medium confidence