VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CRYPTOGRAPHY AND NETWORK SECURITY

LAB 2 : WRITE A PROGRAM IN SUITABLE PROGRAMMING LANGUAGE FOR THE FOLLOWING:

- ➔ 1. VIGENERE CIPHER
- ➔ 2. RAIL FENCE CIPHER
- ➔ 3. EUCLIDEAN ALGORITHM

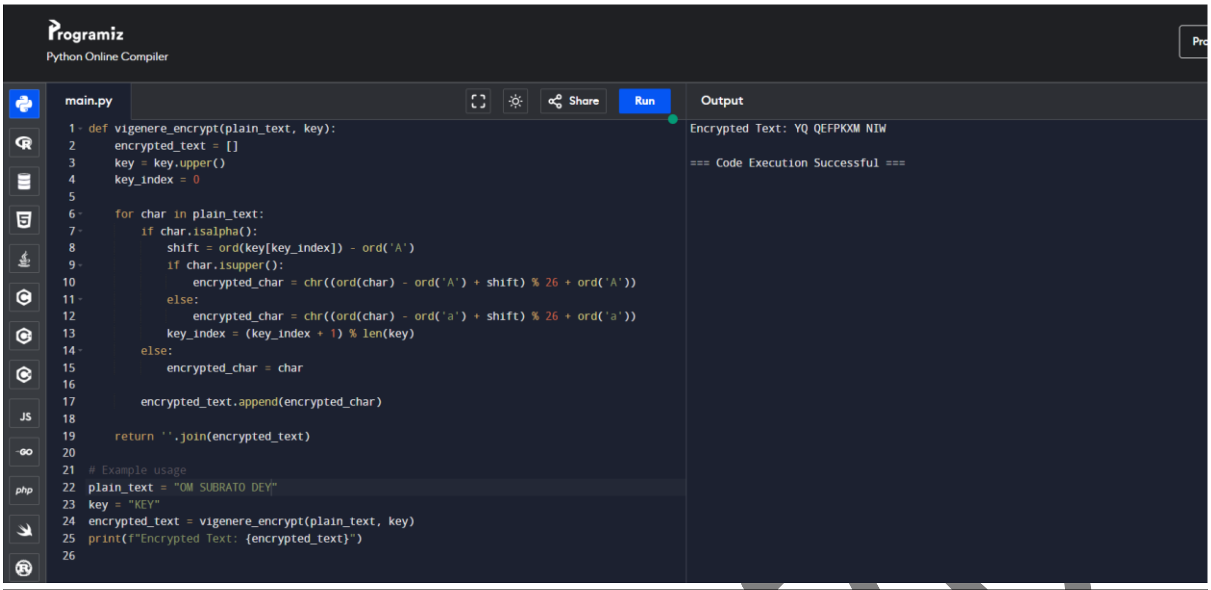
NAME : OM SUBRATO DEY

REGISTER NUMBER : 21BAI1876

1. VIGENERE CIPHER:

ENCRYPTION CODE AND OUTPUT:

```
def vigenere_encrypt(plain_text, key):
    encrypted_text = []
    key = key.upper()
    key_index = 0

    for char in plain_text:
        if char.isalpha():
            shift = ord(key[key_index]) - ord('A')
            if char.isupper():
                encrypted_char = chr((ord(char) - ord('A') + shift)
% 26 + ord('A'))
            else:
                encrypted_char = chr((ord(char) - ord('a') + shift)
% 26 + ord('a'))
            key_index = (key_index + 1) % len(key)
        else:
            encrypted_char = char

        encrypted_text.append(encrypted_char)

    return ''.join(encrypted_text)

# Example usage
plain_text = "OM SUBRATO DEY"
key = "KEY"
encrypted_text = vigenere_encrypt(plain_text, key)
print(f"Encrypted Text: {encrypted_text}")
```

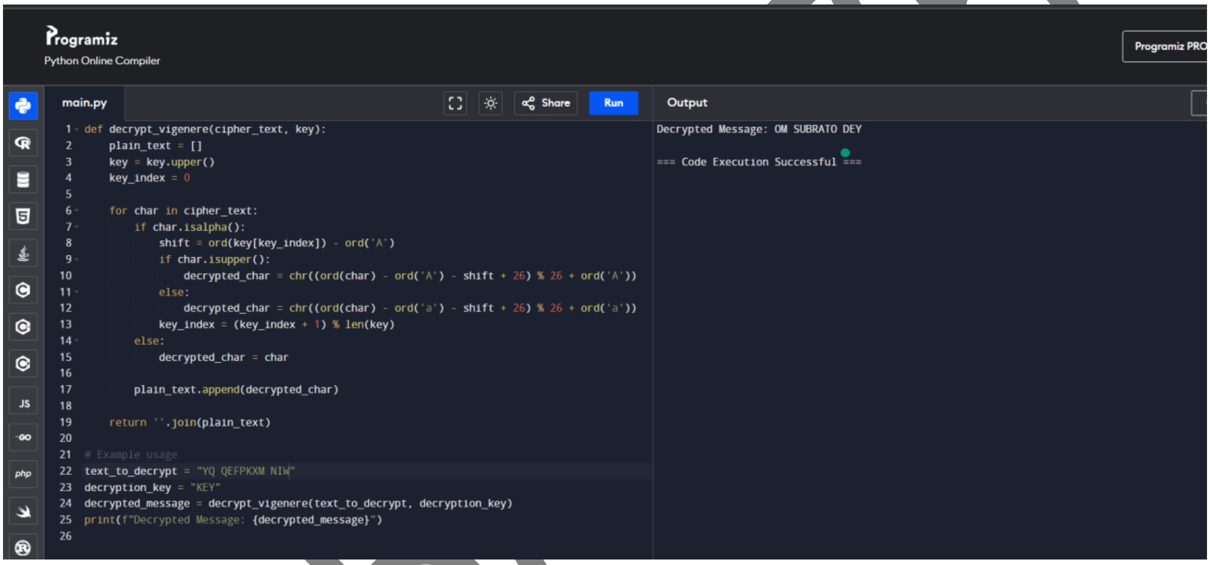


```python
def vigenere_encrypt(plain_text, key):
    encrypted_text = []
    key = key.upper()
    key_index = 0

    for char in plain_text:
        if char.isalpha():
            shift = ord(key[key_index]) - ord('A')
            if char.isupper():
                encrypted_char = chr((ord(char) - ord('A') + shift) % 26 + ord('A'))
            else:
                encrypted_char = chr((ord(char) - ord('a') + shift) % 26 + ord('a'))
            key_index = (key_index + 1) % len(key)
        else:
            encrypted_char = char

        encrypted_text.append(encrypted_char)

    return ''.join(encrypted_text)

# Example usage
plain_text = "OM SUBRATO DEY"
key = "KEY"
encrypted_text = vigenere_encrypt(plain_text, key)
print(f"Encrypted Text: {encrypted_text}")
```

```
Encrypted Text: YQ QEFPKXM NIW

=== Code Execution Successful ===
```

DECRYPTION CODE AND OUTPUT:

```python
def decrypt_vigenere(cipher_text, key):
    plain_text = []
    key = key.upper()
    key_index = 0

    for char in cipher_text:
        if char.isalpha():
            shift = ord(key[key_index]) - ord('A')
            if char.isupper():
                decrypted_char = chr((ord(char) - ord('A') - shift + 26) % 26 + ord('A'))
            else:
                decrypted_char = chr((ord(char) - ord('a') - shift + 26) % 26 + ord('a'))
            key_index = (key_index + 1) % len(key)
        else:
            decrypted_char = char
        plain_text.append(decrypted_char)
    return ''.join(plain_text)


# Example usage
text_to_decrypt = "YQ QEFPKXM NIW"
decryption_key = "KEY"
decrypted_message = decrypt_vigenere(text_to_decrypt,
decryption_key)
print(f"Decrypted Message: {decrypted_message}")
```



```
def decrypt_vigenere(cipher_text, key):
    plain_text = []
    key = key.upper()
    key_index = 0

    for char in cipher_text:
        if char.isalpha():
            shift = ord(key[key_index]) - ord('A')
            if char.isupper():
                decrypted_char = chr((ord(char) - ord('A') - shift + 26) % 26 + ord('A'))
            else:
                decrypted_char = chr((ord(char) - ord('a') - shift + 26) % 26 + ord('a'))
            key_index = (key_index + 1) % len(key)
        else:
            decrypted_char = char

        plain_text.append(decrypted_char)

    return ''.join(plain_text)

# Example usage
text_to_decrypt = "YQ QEFPKXM NIW"
decryption_key = "KEY"
decrypted_message = decrypt_vigenere(text_to_decrypt, decryption_key)
print(f"Decrypted Message: {decrypted_message}")
```

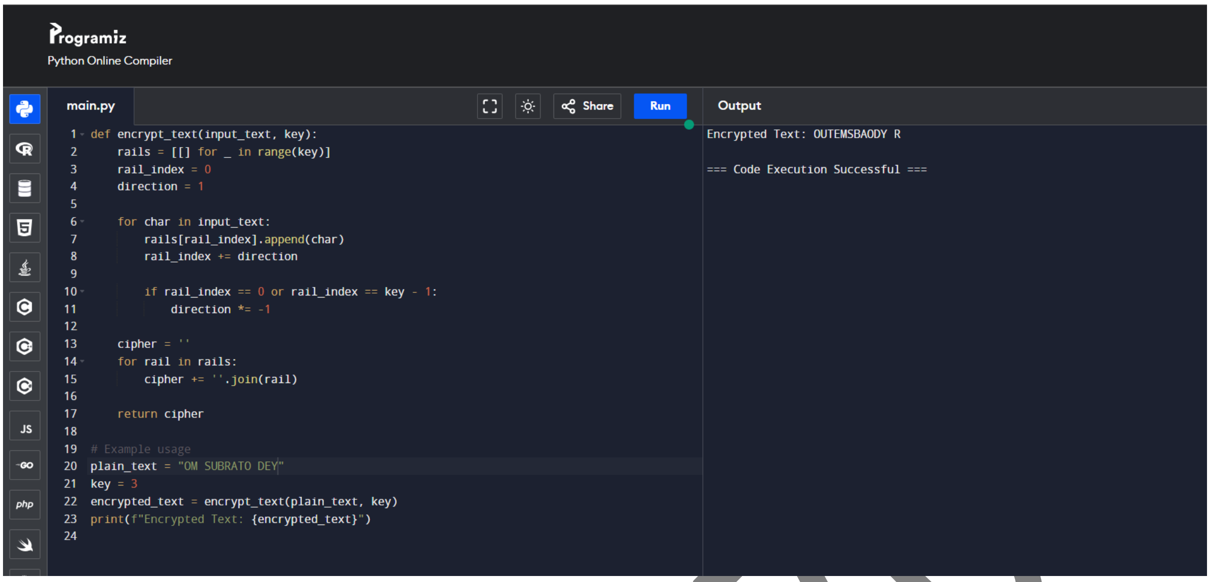
```
Decrypted Message: OM SUBRATO DEY

=== Code Execution Successful ===
```

2. RAIL FENCE CIPHER:

CODE:

```
def encrypt_text(input_text, key):
    rails = [[] for _ in range(key)]
    rail_index = 0
    direction = 1

    for char in input_text:
        rails[rail_index].append(char)
        rail_index += direction

        if rail_index == 0 or rail_index == key - 1:
            direction *= -1

    cipher = ''
    for rail in rails:
        cipher += ''.join(rail)

    return cipher

# Example usage
plain_text = "OM SUBRATO DEY"
key = 3
encrypted_text = encrypt_text(plain_text, key)
print(f"Encrypted Text: {encrypted_text}")
```

OUTPUT:



```python
def encrypt_text(input_text, key):
    rails = [[] for _ in range(key)]
    rail_index = 0
    direction = 1

    for char in input_text:
        rails[rail_index].append(char)
        rail_index += direction

        if rail_index == 0 or rail_index == key - 1:
            direction *= -1

    cipher = ''
    for rail in rails:
        cipher += ''.join(rail)

    return cipher

# Example usage
plain_text = "OM SUBRATO DEY"
key = 3
encrypted_text = encrypt_text(plain_text, key)
print(f"Encrypted Text: {encrypted_text}")
```

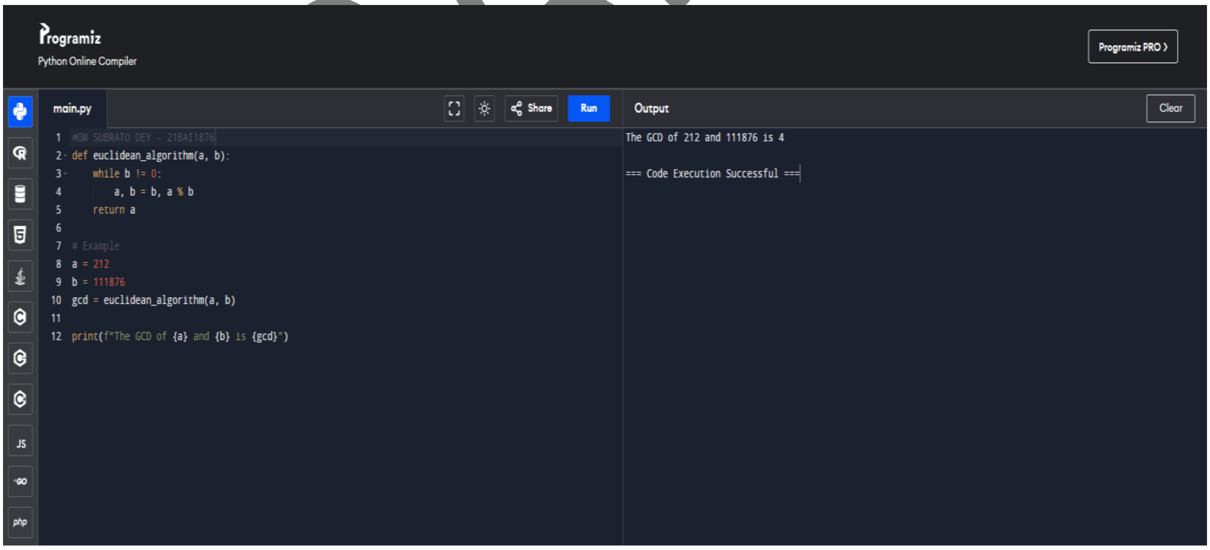
```
Encrypted Text: OUTEMSBAODY R

=== Code Execution Successful ===
```

3. EUCLIDEAN TECHNIQUE:

CODE:

```
def euclidean_algorithm(a, b):
    while b != 0:
        a, b = b, a % b
    return a

# Example
a = 212
b = 111876
gcd = euclidean_algorithm(a, b)

print(f"The GCD of {a} and {b} is {gcd}")
```

OUTPUT:



```
#OM SUBRATO DEY - 21BAI1876
def euclidean_algorithm(a, b):
    while b != 0:
        a, b = b, a % b
    return a

# Example
a = 212
b = 111876
gcd = euclidean_algorithm(a, b)

print(f"The GCD of {a} and {b} is {gcd}")
```

```
The GCD of 212 and 111876 is 4

=== Code Execution Successful ===
```