Name - Khushi Chhatwani
Roll no. csc/21/55
Course -B.Sc(H) CS

Ques 11. Implement a stream cipher technique.

```
def rc4_keystream(key):
"""Generate a pseudorandom keystream using the RC4 algorithm."""
S = list(range(256))
j = 0
for i in range(256):
j = (j + S[i] + key[i % len(key)]) % 256
S[i], S[j] = S[j], S[i]
i = 0
j = 0
while True:
i = (i + 1) % 256
j = (j + S[i]) % 256
S[i], S[j] = S[j], S[i]
yield S[(S[i] + S[j]) % 256]
def stream_cipher(plaintext, key):
keystream = rc4_keystream(key)
ciphertext = []
for byte in plaintext:
keystream_byte = next(keystream)
ciphertext_byte = byte ^ keystream_byte
ciphertext.append(ciphertext_byte)
return bytes(ciphertext)
if __name__ == "__main__":
plaintext = b"do not reply to this mail"
key = b"secretkey"
ciphertext = stream_cipher(plaintext, key)
print("Cipher Text ==> ",ciphertext)
decrypted_plaintext = stream_cipher(ciphertext, key)
print("Deciphered Text ==> ",decrypted_plaintext)
```

Output

```
def rc4_keystream(key):
    """Generate a pseudorandom keystream using the RC4 algorithm."""
    S = list(range(256))
    j = 0
    for i in range(256):
        j = (j + S[i] + key[i % len(key)]) % 256
        S[i], S[j] = S[j], S[i]
    i = 0
    j = 0
    while True:
        i = (i + 1) % 256
        j = (j + S[i]) % 256
        S[i], S[j] = S[j], S[i]
        yield S[(S[i] + S[j]) % 256]

def stream_cipher(plaintext, key):
    keystream = rc4_keystream(key)
    ciphertext = []
    for byte in plaintext:
        keystream_byte = next(keystream)
        ciphertext_byte = byte ^ keystream_byte
        ciphertext.append(ciphertext_byte)
    return bytes(ciphertext)

if __name__ == "__main__":
    plaintext = b"do not reply to this mail"
    key = b"secretkey"
    ciphertext = stream_cipher(plaintext, key)
    print("Cipher Text ==> ",ciphertext)
    decrypted_plaintext = stream_cipher(ciphertext, key)
    print("Deciphered Text ==> ",decrypted_plaintext)
```

```
__name__ == "__main__":
plaintext = b"do not reply to this mail"
key = b"secretkey"
ciphertext = stream_cipher(plaintext, key)
print("Cipher Text ==> ",ciphertext)
decrypted_plaintext = stream_cipher(ciphertext, key)
print("Deciphered Text ==> ",decrypted_plaintext)
```

```
her Text ==>  b"\xc1\xc7\xfcr#\xc5\xb5:M\xdc'\xfd\xa2S[\x0b\x00?'\x13\xbe\x1eO]\xf1"
iphered Text ==>  b'do not reply to this mail'
```