jupyter Untitled1 Last Checkpoint: 11 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [1]:
```
import random
```

In [2]:
```
def isPrime(n):
    if n==0 or n==1:
        return False

    r = int(n/2)

    for i in range(2, r):
        if(n%i == 0):
            return False

    return True
```

In [3]:
```
def generatePrimes():
    primes = [i for i in range(1, 999) if isPrime(i)]

    return random.choices(primes, k=2)
```

In [4]:
```
class RSA:
    def __init__(self, p, q):
        self.p = p
        self.q = q
        self.N = p * q
        self.product = (p-1) * (q-1)
        self.generateKeys()

    def generateKeys(self):

        for i in range(1, 999999):
            if(self.product % i != 0):
                self.E = i
                break

        for i in range(1, self.product-1):
            if((i*self.E) % self.product == 1):
                self.D = i
                break

        print("Encryption Key (N, E): (", self.N, ", ", self.E, ")")
        print("Decryption Key (N, D): (", self.N, ", ", self.D, ")")


    def encrypt(self, plaintext):

        pt = []
        ct = []

        for i in plaintext:
            pt.append(ord(i))

        for i in pt:
            ct.append((i**self.E)%self.N)

        return ct

    def decrypt(self, ciphertext):
        dt = []

        for i in ciphertext:
            dt.append(chr((i**self.D)%self.N))

        return ''.join(dt)


    def encrypt_2(self, plaintext):
        ct = (int(plaintext)**self.E) % self.N

        return ct

    def decrypt_2(self, ciphertext):
        dt = (int(ciphertext)**self.D) % self.N

        return dt
```

In [9]:
```
if __name__ == "__main__":

    p, q = generatePrimes()
#     print("Generated Primes are:\nP = ", p, "\nQ = ", q)

    p = int(input("Enter P: "))
    q = int(input("Enter Q: "))

    print("----------------------------------")

    rsa = RSA(p, q)

    print("----------------------------------")

    pt = input("Enter the Plaintext: ")

    print("----------------------------------")

    ct = rsa.encrypt(pt)

    print("Encrypted Ciphertext : ", ct)

    dt = rsa.decrypt(ct)

    print("Decrypted Text: ", dt)


    print("----------------------------------")

    ct2 = rsa.encrypt_2(pt)

    print("Encrypted Ciphertext : ", ct2)
```

```
    dt2 = rsa.decrypt_2(ct2)

    print("Decrypted Text: ", dt2)
```

```
Enter P: 17
Enter Q: 29
---------------------------------
Encryption Key (N, E): ( 493 ,  3 )
Decryption Key (N, D): ( 493 ,  299 )
---------------------------------
Enter the Plaintext: 10
---------------------------------
Encrypted Ciphertext :  [315, 160]
Decrypted Text:  10
---------------------------------
Encrypted Ciphertext :  14
Decrypted Text:  10
```

In [ ]: