Implementation of RSA Algorithm

```
import random
import ast


def isPrime(n):
    if n == 0 or n == 1:
        return False

    for i in range(2,int(n**0.5)+1):
        if n%i==0:
            return False

    return True

def generate_primes():
    primes = [ i for i in range(0, 999) if isPrime(i) ]
    return random.choices(primes, k=2)


class RSA :

    def __init__(self, p, q):
        self.p = p
        self.q = q
        self.N = p * q
        self.product = (p - 1) * (q - 1)
        self.generate_keys()
    # (N, E) (N, D)
    def generate_keys(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 : {}'.format(self.E))
        print('Decryption Key : {}'.format(self.D))

    def encrypt(self, text) :
        pt = []
        ct = []
        for i in text:
            pt.append(ord(i))

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

        return ct

    def decrypt(self, cipher) :
        dt = []
        for i in cipher:
            dt.append(chr(((i ** self.D) % self.N)))

        return ''.join(dt)


if __name__ ==  "__main__" :
    p, q = generate_primes()

    print('Generated Primes are p = {}, q = {}'.format(p,q))
    rsa = RSA(p, q)

    text = input('Enter text to encrypt : ')

    ct = rsa.encrypt(text)

    print('Encrypted text : {}'.format(ct))

    decrypted_text = rsa.decrypt(ct)

    print('Descrypted Message : {}'.format(decrypted_text))

     Generated Primes are p = 461, q = 953
     Encryption Key : 3
     Decryption Key : 291947
     Enter text to encrypt : hello
     Encrypted text : [246198, 151635, 381046, 381046, 49632]
     Descrypted Message : hello
```

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.