**Final Year B.Tech. (CSE) – II [ 2022-23 ]**

**Cryptograpy and Network Security Lab**

**PRN: 2019BTECS00015**

**Full name: Shraddha Sanjay Kharat**

**Batch: B1**

**Assignment No. 9**

**Title:**

Prime Factorization

**Aim:**

To Demonstrate Prime Factorization

**Theory:**

RSA Laboratories states that: for each RSA number n, there exists prime numbers p and q such that

n = p × q.

The problem is to find these two primes, given only n.

**Code:**

from sympy.ntheory import factorint

import math

def factors\_int(num):

    poss\_p = math.floor(math.sqrt(num))

    if poss\_p % 2 == 0:

        poss\_p += 1

    while poss\_p < num:

        if num % poss\_p == 0:

            return poss\_p

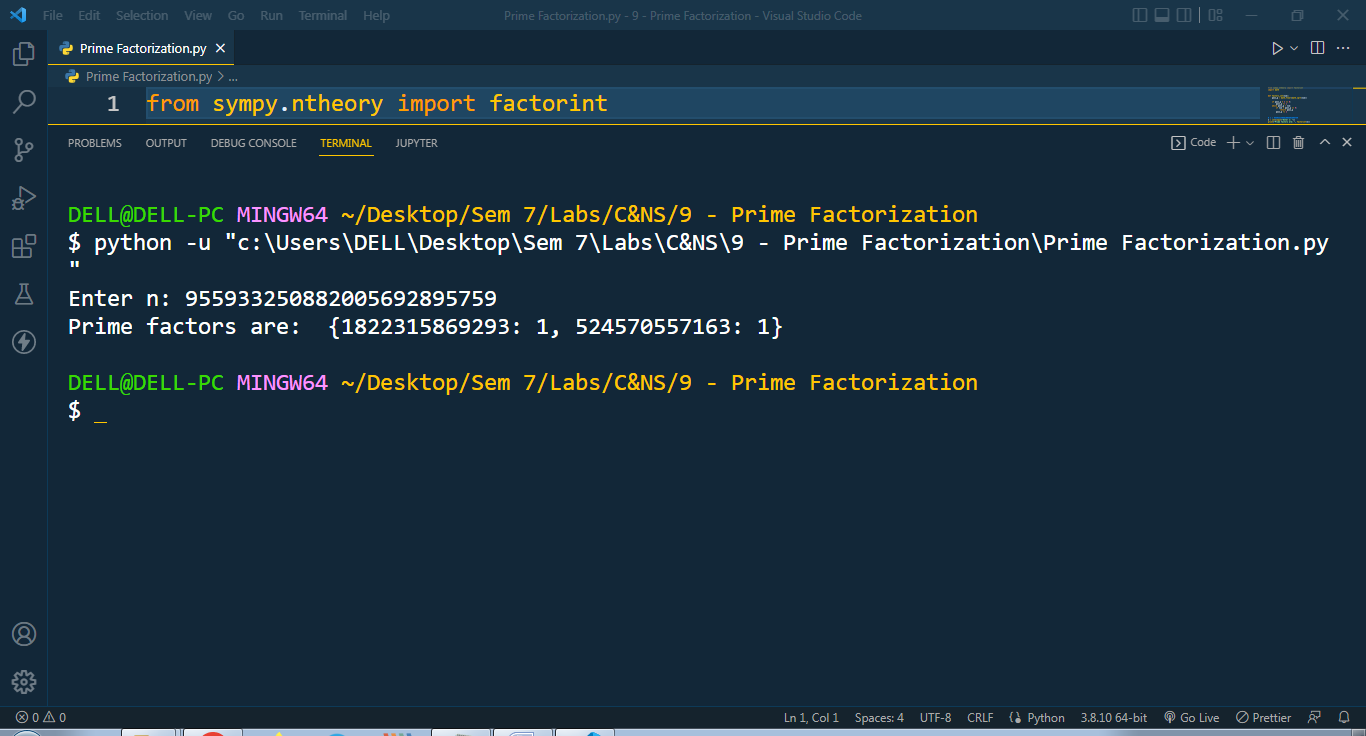
        poss\_p += 2

*# n = 955933250882005692895759*

n = int(input("Enter n: "))

print(factorint(n))

**Output:**

****

**Conclusion:**

**The RSA Factoring Challenge was a challenge put forward by RSA Laboratories to encourage research into computational number theory and the practical difficulty of factoring large integers and cracking RSA keys used in cryptography. They published a list of semiprimes (numbers with exactly two prime factors) known as the RSA numbers, with a cash prize for the successful factorization of some of them.**