Cryptography & Network Security

PRN - 2019BTECS00026

Name - Niraja Vasudev Kulkarni

Batch - B1

Assignment - 8

Title: Euclidean and Extended Euclidean Algorithm

Aim: To Demonstrate Euclidean and Extended Euclidean Algorithm

Theory:

In mathematics, the Euclidean algorithm, or Euclid's algorithm, is an efficient method for computing the greatest common divisor (GCD) of two integers (numbers), the largest number that divides them both without a remainder.

The extended Euclidean algorithm is particularly useful when a and b are coprime. With that provision, x is the modular multiplicative inverse of a modulo b, and y is the modular multiplicative inverse of b modulo a.

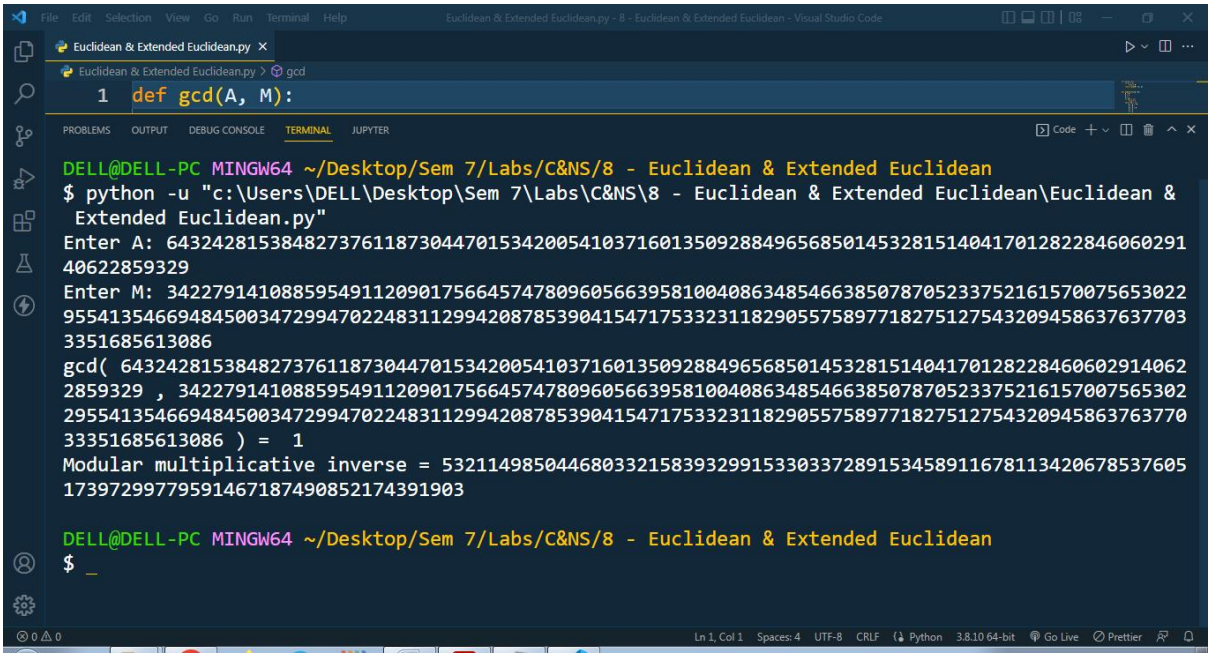
Code:

```
def gcd(A, M):
  if A == 0:
    return M
  return gcd(M % A, A)


def modInverse(A, M):
  m = M
  y = 0
```

```
    x = 1
    if (M == 1):
        return 0
    while (A > 1):
        q = A // M
        t = M
        M = A % M
        A = t
        t = y
        y = x - q * y
        x = t
    if (x < 0):
        x = x + m
    return x

# A =
643242815384827376118730447015342005410371601350928849656850145328151404170128228460602914062285932 9
# X =
532114985044680332158393299153303372891534589116781134206785376051739729977959146718749085217439190 3
# M =
342279141088595491120901756645747809605663958100408634854663850787052337521615700756530229554135466948450034729947022483112994208785390415471753323118290557589771827512754320945863763770333516856130 86

# A * X = 1 mod M
A = int(input("Enter A: "))
M = int(input("Enter M: "))
```

```
print("gcd(", A, ",", M, ") = ", gcd(A, M))

print("Modular multiplicative inverse =", modInverse(A, M))
```

Output:



Conclusion:

The Euclidean and Extended Euclidean algorithm are used to find the GCD of numbers and the Multiplicative inverse of two coprime numbers respectively.