**Cryptography & Network Security**

**PRN - 2019BTECS00026**

**Name - Niraja Vasudev Kulkarni**

**Batch - B1**

**Assignment - 10**

**Title**: Chinese Remainder Theorem

**Aim:** To Demonstrate Chinese Remainder Theorem

**Theory:**

In mathematics, the Chinese remainder theorem states that if one knows the remainders of the Euclidean division of an integer n by several integers, then one can determine uniquely the remainder of the division of n by the product of these integers, under the condition that the divisors are pair wise co-prime.

**Code:**

def Extended(x, m):

    r1 = m

    r2 = x

    t1 = 0

    t2 = 1

    while(r2 > 0):

        q = r1 // r2

        r = r1 % r2

        t = t1 - q \* t2

        r1 = r2

        r2 = r

        t1 = t2

        t2 = t

    if(t1 < 0):

        return t1 + m

    return t1

def findMinX(num, rem, k):

    prod = 1

    for i in range(0, k):

        prod = prod \* rem[i]

    result = 0

    for i in range(0, k):

        pp = prod // rem[i]

        result = result + num[i] \* Extended(pp, rem[i]) \* pp

    return result % prod

# num = [129934811447123020117172145698449, 129934811447123020117172145698449]

# rem = [25, 4]

# x = 129934811447123020117172145698449(mod 25)

# x = 129934811447123020117172145698449(mod 4)

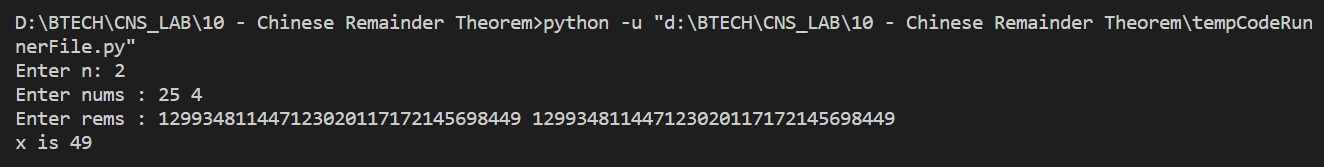
n = int(input("Enter n: "))

num = list(map(int, input("Enter nums : ").strip().split()))[:n]

rem = list(map(int, input("Enter rems : ").strip().split()))[:n]

print("x is", findMinX(num, rem, n))

**Output:**



**Conclusion:**

The Chinese remainder theorem is widely used for computing with large integers, as it allows replacing a computation for which one knows a bound on the size of the result by several similar computations on small integers.