

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 inv(a, m):  
    m0 = m  
    x0 = 0  
    x1 = 1  
  
    if (m == 1):  
        return 0  
  
    while (a > 1):  
        q = a // m  
        t = m
```

```

    m = a % m

    a = t

    t = x0

    x0 = x1 - q * x0

    x1 = t

if (x1 < 0):
    x1 = x1 + m0

return x1

def findMinX(num, rem, k):
    prod = 1
    for i in range(0, k):
        prod = prod * num[i]

    result = 0

    for i in range(0, k):
        pp = prod // num[i]
        result = result + rem[i] * inv(pp, num[i]) * pp

    return result % prod

# num = [25, 4]
# rem = [129934811447123020117172145698449, 129934811447123020117172145698449]
# x = 129934811447123020117172145698449(mod 25)
# x = 129934811447123020117172145698449(mod 4)
n = int(input("Enter n: "))

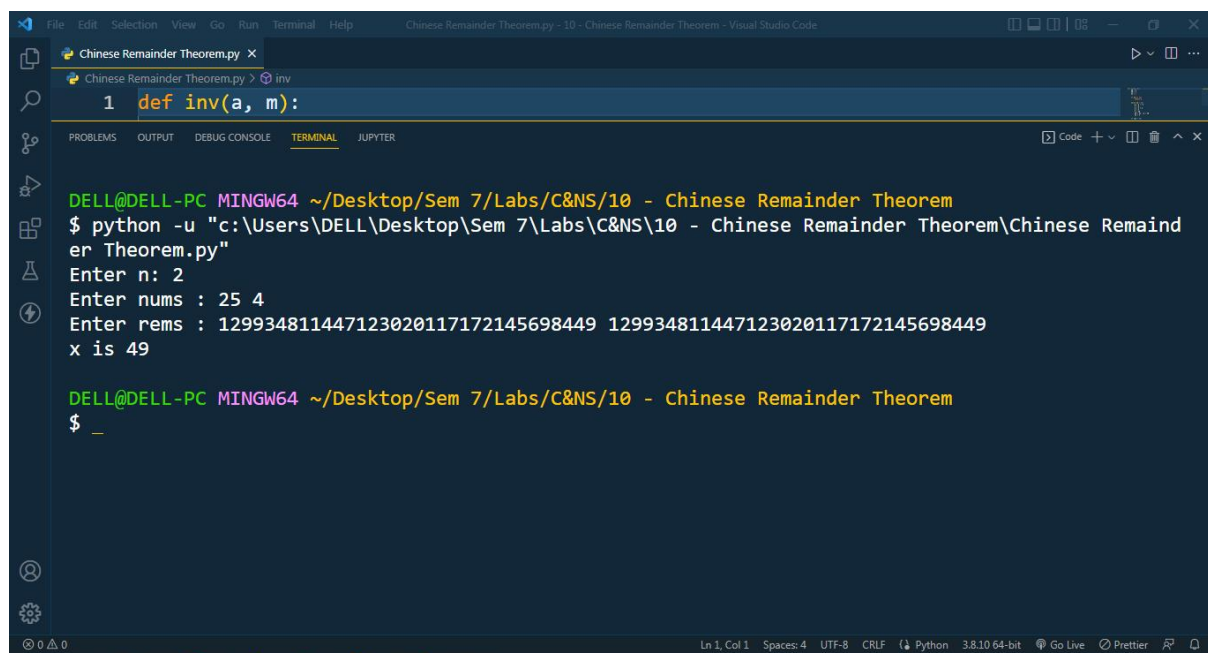
```

```
rem = []

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:



The screenshot shows a Visual Studio Code window with a file named 'Chinese Remainder Theorem.py'. The code editor displays a function definition: `def inv(a, m):`. Below the editor, the 'TERMINAL' tab is active, showing the command prompt output. The user has run the command `python -u "c:\Users\DELL\Desktop\Sem 7\Labs\C&NS\10 - Chinese Remainder Theorem\Chinese Remainder Theorem.py"`. The program prompts for 'Enter n: 2', 'Enter nums : 25 4', and 'Enter rems : 129934811447123020117172145698449 129934811447123020117172145698449'. The final output is 'x is 49'.

```
DELL@DELL-PC MINGW64 ~/Desktop/Sem 7/Labs/C&NS/10 - Chinese Remainder Theorem
$ python -u "c:\Users\DELL\Desktop\Sem 7\Labs\C&NS\10 - Chinese Remainder Theorem\Chinese Remainder Theorem.py"
Enter n: 2
Enter nums : 25 4
Enter rems : 129934811447123020117172145698449 129934811447123020117172145698449
x is 49

DELL@DELL-PC MINGW64 ~/Desktop/Sem 7/Labs/C&NS/10 - Chinese Remainder Theorem
$ _
```

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.