Date:11.11.2022

Final Year B. Tech., Sem VII 2022-23

Cryptography And Network Security Lab

Assignment submission

PRN No: 2019BTECS00077

Full name: Avinash Vishnu Biradar

Batch: B7

Assignment: 9

Title of assignment: Implementation of Prime Factorization of Semiprime Number

Title:

Implementation of Prime Factorization of Semiprime Number

Aim:

To develop and implement Prime Factorization of Semiprime Number

Theory:

- In mathematics, a semiprime is a natural number that is the product of exactly two prime numbers.
- The two primes in the product may equal each other, so the semiprimes include the squares of prime numbers. Because there are infinitely many prime numbers, there are also infinitely many semiprimes.
- Semiprimes are also called biprimes.

- Semiprimes that are not square numbers are called discrete, distinct, or squarefree semiprimes.
- The semiprimes less than 100 are:
  4, 6, 9, 10, 14, 15, 21, 22, 25, 26, 33, 34, 35, 38, 39, 46, 49, 51, 55, 57, 58, 62, 65, 69, 74, 77, 82, 85, 86, 87, 91, 93, 94, and 95
- A semiprime counting formula was discovered by E. Noel and G. Panos in 2005. Let $Pi_2(n)$ denote the number of semiprimes less than or equal to n. Then

  $$\pi_2(n) = \sum_{k=1}^{\pi(\sqrt{n})} [\pi(n/p_k) - k + 1]$$

  where Pi(x) is the prime-counting function and p_k denotes the kth prime.

```
Enter the Number:
3595
**
5
719



--------------------------------
Process exited after 106.9 seconds with return value 0
Press any key to continue . . .
```

Python Code:

```
import math
import time

def getFactors(A,n):
   print("Factorizing.....")
   end = (int)(math.sqrt(A)) + 1;
   start_time = time.time()
   for i in range(2, end):
      if A % i == 0:
```

```
            print(f"Factors --> {i} , {A // i}")
            break
    end_time = time.time()
    print("Number of Digits: ", n)
    print("Execution Time: ", end_time - start_time)
    print("")


semiprimes = [1000010666663, 10000079888899, 100000008888889,
1000000518888883, 10000000058888887, 100000005178888819,
1000000000099987889, 10000000000097987989, 100000000088787877679]
digits = [12, 13, 14, 15, 16, 18, 17, 19, 20]


for i in range(0,len(semiprimes)):
    getFactors(semiprimes[i],digits[i])
```

Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

G:\Kunal\Sem-7\C&NS Assignments\Experiment - 9>python3 factorizationChallenge.py
Python was not found; run without arguments to install from the Microsoft Store,
or disable this shortcut from Settings > Manage App Execution Aliases.

G:\Kunal\Sem-7\C&NS Assignments\Experiment - 9>python factorizationChallenge.py
Factorizing.....
Factors --> 333337 , 2999999
Number of Digits:  12
Execution Time:  0.06249737739562988

Factorizing.....
Factors --> 989999 , 10101101
Number of Digits:  13
Execution Time:  0.18745851516723633

Factorizing.....
Factors --> 99989 , 1000110101
Number of Digits:  14
Execution Time:  0.015615701675415039

Factorizing.....
Factors --> 11111117 , 89999999
Number of Digits:  15
Execution Time:  1.952669382095337
```

```
C:\Windows\System32\cmd.exe
Factorizing.....
Factors --> 89999999 , 111111113
Number of Digits:  16
Execution Time:  16.094754934310913

Factorizing.....
Factors --> 89999999 , 1111111181
Number of Digits:  18
Execution Time:  15.462402582168579

Factorizing.....
Factors --> 99999989 , 10000001101
Number of Digits:  17
Execution Time:  25.70964503288269

Factorizing.....
Factors --> 99998999 , 100001001011
Number of Digits:  19
Execution Time:  33.65650963783264

Factorizing.....
Factors --> 9999889 , 10000111010111
Number of Digits:  20
Execution Time:  3.108642101287842


G:\Kunal\Sem-7\C&NS Assignments\Experiment - 9>
```

Conclusion:

Performed the experiment successfully.

The program done above is for limited numbers till range of 10000 but there are codes which can factorize the numbers having 20-25 digits of range.