PRIME NUMBER GENERATOR AND CHECKER

Name:- Aman Sarkar

Roll No. :- 202401100400027

Course :- BTech

Branch :- CSE(AI&ML)

Course Code :- AI101B

Introduction:-

Prime Number Generator and Checker - Problem Summary

The task involves two main functions:

1. **Prime Number Generator: Generate all prime numbers within a given range [a, b].**
2. **Prime Number Checker: Verify whether a given number n is prime.**

Constraints

- **1 ≤ a ≤ b ≤ 10^6 (for prime generation).**
- **1 ≤ n ≤ 10^6 (for prime checking).**

Example:-

```
Generated Prime Numbers:
    Prime Numbers
0               2
1               3
2               5
3               7
4              11
5              13
6              17
7              19
8              23
9              29
10             31
11             37
12             41
13             43
14             47
15             53
16             59
17             61
18             67
19             71
```

Methodology:-

1. **Prime Number Generator: Use the Sieve of Eratosthenes to mark non-prime numbers efficiently.**
2. **Start with an array assuming all numbers are prime and eliminate multiples of each prime.**
3. **Extract the remaining numbers in the given range [a, b] as prime numbers.**
4. **Prime Number Checker: Use Trial Division to check divisibility up to √n.**
5. **If n is divisible by any number in this range, it is not prime; otherwise, it is prime.**
6. **The generator runs in O(n log log n), while the checker runs in O(√n), ensuring efficiency.**
7. **The approach is optimized for handling large inputs up to 10^6.**

CODE:-

```python
import pandas as pd
import matplotlib.pyplot as plt

# Function to check if a number is prime
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

# Function to generate prime numbers up to a given limit
def generate_prime_numbers(limit):
    return [n for n in range(limit) if is_prime(n)]

# Function to print and visualize prime numbers
def prime_generator_and_checker(limit):
    primes = generate_prime_numbers(limit)
    df = pd.DataFrame({"Prime Numbers": primes})
    print("Generated Prime Numbers:")
    print(df.head(20))  # Display first 20 primes

    # Plot
    plt.figure(figsize=(10, 5))
    plt.scatter(df.index, df["Prime Numbers"], color="blue", s=10, label="Prime Numbers")
    plt.xlabel("Index")
    plt.ylabel("Prime Numbers")
    plt.title("Prime Number Distribution")
    plt.legend()
    plt.grid()
    plt.show()

# Example usage
limit = 100  # Change this limit as needed
prime_generator_and_checker(limit)
```

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```python
df = pd.read_csv('/content/drive/MyDrive/prime_numbers (1).csv')
```

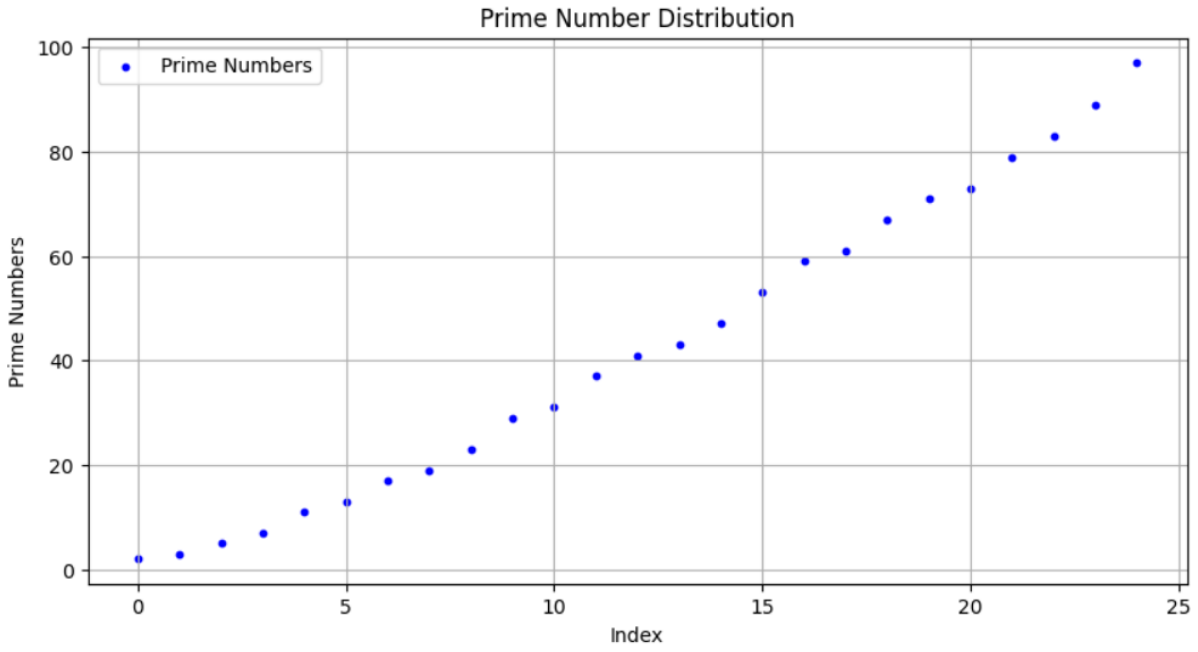
```python
print(df.head())
```

```
   Index  Prime Numbers
0      1              2
1      2              3
2      3              5
3      4              7
4      5             11
```

```
[8] print(df.shape)

(168, 2)

[9] print(df.describe())

             Index  Prime Numbers
count  168.000000     168.000000
mean    84.500000     453.136905
std     48.641546     298.192417
min      1.000000       2.000000
25%     42.750000     188.500000
50%     84.500000     436.000000
75%    126.250000     703.000000
max    168.000000     997.000000

print(df.dtypes)

Index           int64
Prime Numbers   int64
dtype: object
```

OUTPUT:-

Generated Prime Numb	
	Prime Numbers
0	2
1	3
2	5
3	7
4	11
5	13
6	17
7	19
8	23
9	29
10	31
11	37
12	41
13	43
14	47
15	53
16	59
17	61
18	67
19	71


Prime Number Distribution
Prime Numbers
Prime Numbers
Index
0
20
40
60
80
100
0
5
10
15
20
25