EXPERIMENT NO: 10 Prime Number Analyzer

AIM: Write a python program to analyze the input number is prime or not

THEORY:

A **prime number** is a natural number greater than **1** that has no positive divisors other than **1** and itself. If a number is **not prime**, it means it has factors other than **1** and itself.

For example:

- **Prime numbers**: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29...
- **Non-prime numbers**: 4, 6, 8, 9, 10, 12...

To check whether a number n is prime, we check whether it is **divisible** by any number from **2 to n/2(or √n for efficiency)**. If any divisor is found, the number is not prime.

Algorithm for Prime Number Checking Program

```
1. Start
2. Input a number.
3. Initialize count = 0 and limit = number // 2.
4. Check if the number is divisible by any number from 2 to limit:
        If number % i == 0, set count = 1 and exit loop.
5. Check the count variable:
        • If count >= 1, print "Not a Prime Number."
        • Otherwise, print "Prime Number."
6. End
```

PROGRAM:

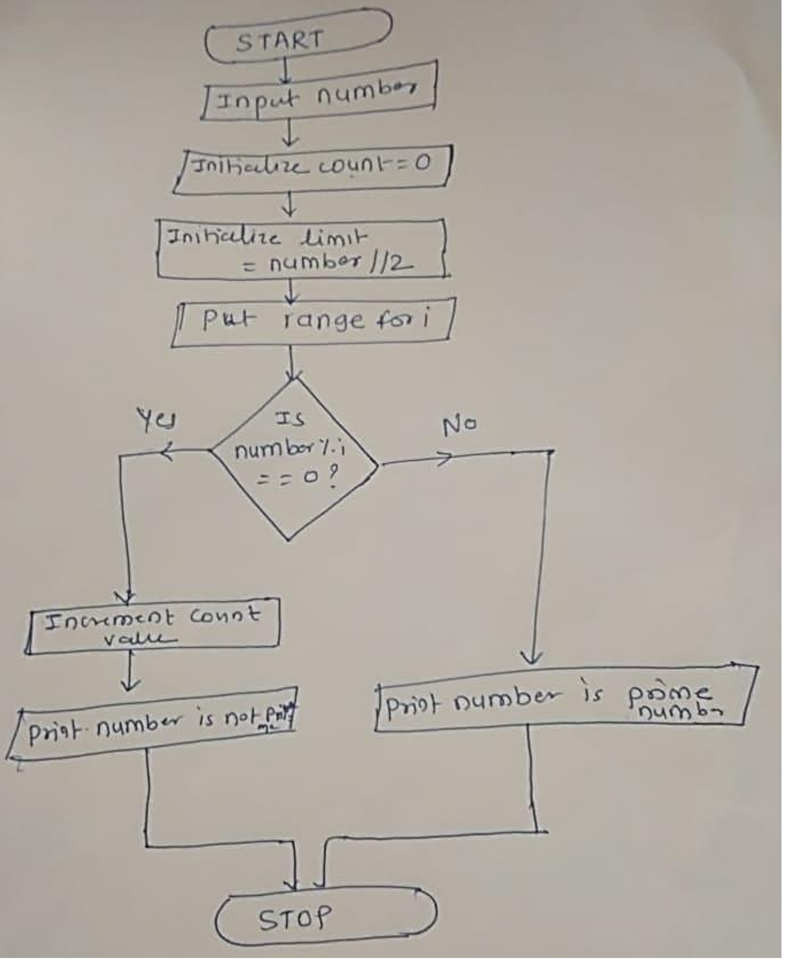
```
def check_prime(number):
    count = 0
    limit = number // 2                 # The largest possible divisor of a number is half of the number.

    # Check for divisors from 2 to number//2
    for i in range(2, limit + 1):
        if number % i == 0:             # If number is divisible by i
            count += 1                  # Increment count because it's divisible by i
            break                       # No need to check further, we know the number is not prime.

    # If count is 1 or more, it's not a prime
    if count >= 1:
        print(f"{number} is not a prime number.")
    else:
        print(f"{number} is a prime number.")

# Function call
check_prime(29)
check_prime(8)
```

Flowchart : To check number is prime or not


START
Input number
Initialize count = 0
Initialize limit = number//2
Put range for i
Is number%i == 0?
Yes
No
Increment count value
Print number is not prime
Print number is prime number
STOP