AI (MSE 1) PROJECT REPORT

TOPIC – Prime Number Generator And Checker

Name – Sanskriti Agrawal

Course – B tech

Branch – CSEAI

Section – C

Univ Roll No – 202401100300216

Class Roll No – 69

INTRODUCTION

Prime Number Generator And Checker

A prime number generator and checker is a useful tool for identifying prime numbers and generating a list of them within a given range. Prime numbers are natural numbers greater than 1 that can only be divided by 1 and themselves.

The provided Python program includes two key functions: is_prime(n), which checks if a number is prime, and generate_primes(limit), which generates a list of prime numbers up to a specified limit. By efficiently determining prime numbers, this program can assist in tasks such as number theory research, encryption algorithms, and mathematical problem-solving.

METHODOLOGY

The methodology behind the prime number generator and checker involves two main functions: is_prime(n) and generate_primes(limit).

1. **Prime Number Checking (is_prime(n))**:
   - The function first checks if n is less than 2, as numbers less than 2 are not prime.
   - It then iterates from 2 to the square root of n, checking if n is divisible by any number in this range.
   - If n is divisible by any of these numbers, it returns False; otherwise, it returns True, confirming n is prime.
2. **Prime Number Generation (generate_primes(limit))**:
   - The function initializes an empty list primes to store prime numbers.
   - It iterates through numbers from 2 to the specified limit and checks each number using the is_prime(n) function.
   - If a number is prime, it is added to the list.
   - Finally, the function returns the list of all prime numbers up to limit.
3. **Implementation & Execution**:
   - The script includes an example usage where it checks if 29 is prime and generates prime numbers up to 50.
   - The results are displayed using print statements.

This approach ensures an efficient and structured method for identifying and listing prime numbers using mathematical principles.

CODE TYPED

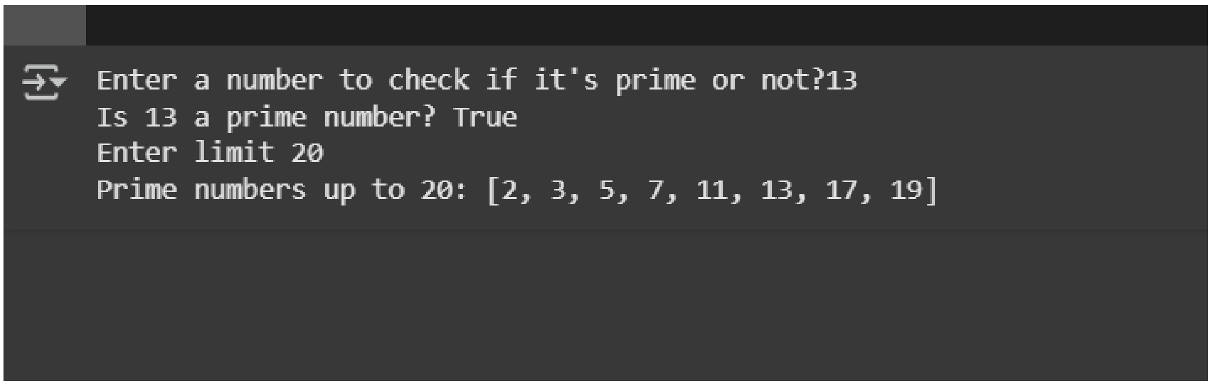
```
def is_prime(n):
    """
    Check if a number is prime.
    :param n: Integer to check
    :return: True if prime, False otherwise
    """
    if n < 2:
        return False
    for i in range(2, int(n ** 0.5) + 1):  # Check divisibility up to sqrt(n)
        if n % i == 0:
            return False
    return True


def generate_primes(limit):
    """
    Generate a list of prime numbers up to a given limit.
    :param limit: Upper bound for prime numbers
    :return: List of prime numbers
    """
    primes = []
    for num in range(2, limit + 1):
        if is_prime(num):  # Use the is_prime function
            primes.append(num)
    return primes


# Example usage
if __name__ == "__main__":
    num_to_check = int(input("Enter a number to check if it's prime or not?"))
    print(f"Is {num_to_check} a prime number? {is_prime(num_to_check)}")


    limit = int(input("Enter limit "))
    print(f"Prime numbers up to {limit}: {generate_primes(limit)}")
```

Screenshot Of Output



```
Enter a number to check if it's prime or not?13
Is 13 a prime number? True
Enter limit 20
Prime numbers up to 20: [2, 3, 5, 7, 11, 13, 17, 19]
```