



# Prime Number Generator and Checker

Student name – Abhishek Singh

Roll no – 202401100300009

Branch-CSE AI

Subject- AI

# Introduction

Prime numbers are fundamental in mathematics and have significant applications in cryptography, computer science, and number theory. A prime number is a natural number greater than 1 that has only two factors: 1 and itself. This report presents a simple Python program that checks whether a given number is prime and generates prime numbers up to a given limit. The implementation is designed to work in Google Colab and is structured across multiple cells for better modularity and execution control.

# Methodology

The program is divided into six cells to allow for a step-by-step execution in Google Colab. The methodology follows these steps:

**Prime Checker Function** – A function is created to check whether a given number is prime by testing divisibility from 2 to the square root of the number.

**User Input for Prime Check** – The user inputs a number to test for primality.

**Prime Checking Execution** – The program verifies and prints whether the number is prime.

**Prime Generator Function** – A function generates all prime numbers up to a given limit.

**User Input for Prime Generation** – The user enters an upper limit for generating prime numbers.

**Output of Prime Numbers** – The program prints the list of prime numbers up to the specified limit.

This step-by-step approach ensures a clear understanding of the logic while allowing for flexible execution in Google Colab.

# Python Code

# Cell 1: Define Prime Number Checker Function

```
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
```

# Cell 2: Take Input for Prime Checking

```
num = int(input("Enter a number to check if it's prime: "))
```

# Cell 3: Check If the Number is Prime

```
if is_prime(num):  
    print(f"{num} is a prime number.")  
else:  
    print(f"{num} is not a prime number.")
```

# Cell 4: Define Prime Number Generator Function

```
def generate_primes(limit):  
    primes = []  
    for num in range(2, limit + 1):
```

```
    if is_prime(num): # Using function from Cell 1
        primes.append(num)
return primes
```

# Cell 5: Take Input for Prime Number Generation

```
limit = int(input("Enter a number to generate prime numbers up to: "))
```

# Cell 6: Generate and Print Prime Numbers

```
print(f"Prime numbers up to {limit}: {generate_primes(limit)}")
```

# Output and Results

## Example 1: Prime Check

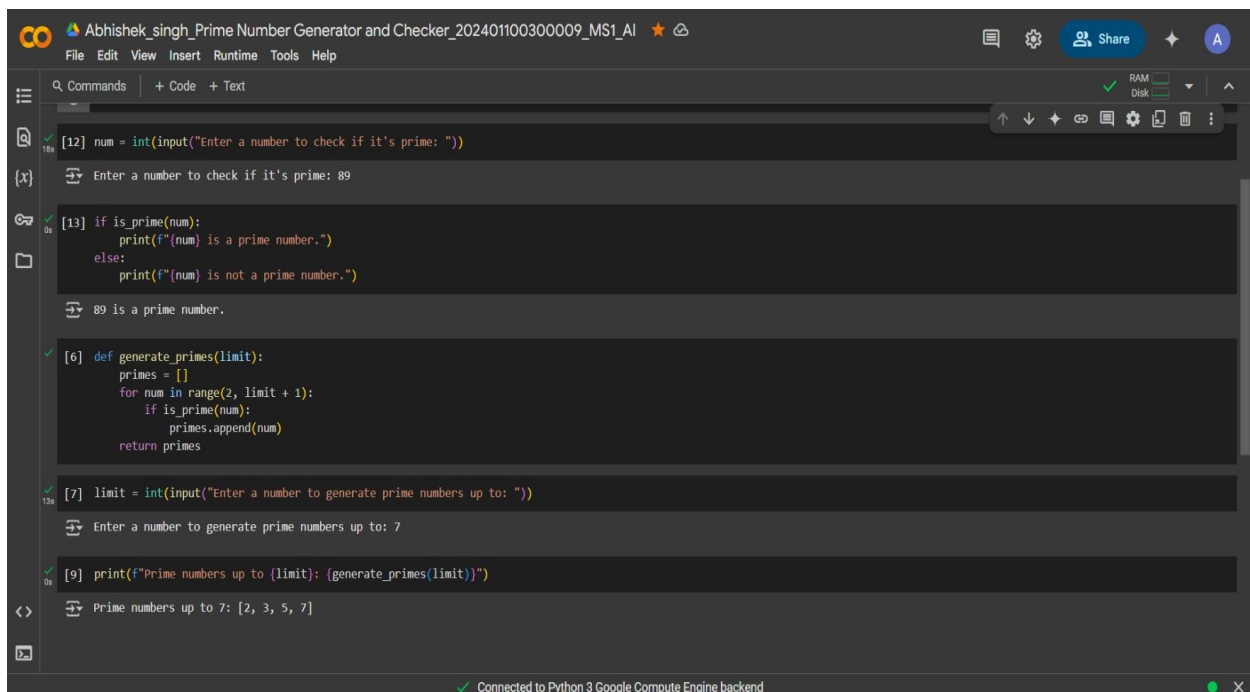
Enter a number to check if it's prime: 7

7 is a prime number.

## Example 2: Prime Generation

Enter a number to generate prime numbers up to: 20

Prime numbers up to 20: [2, 3, 5, 7, 11, 13, 17, 19]



The screenshot shows a Jupyter Notebook titled "Abhishek\_singh\_Prime Number Generator and Checker\_202401100300009\_MS1\_AI". The notebook contains the following code cells:

```
[12] num = int(input("Enter a number to check if it's prime: "))
Enter a number to check if it's prime: 89

[13] if is_prime(num):
    print(f"{num} is a prime number.")
else:
    print(f"{num} is not a prime number.")
89 is a prime number.

[6] def generate_primes(limit):
    primes = []
    for num in range(2, limit + 1):
        if is_prime(num):
            primes.append(num)
    return primes

[7] limit = int(input("Enter a number to generate prime numbers up to: "))
Enter a number to generate prime numbers up to: 7

[9] print(f"Prime numbers up to {limit}: {generate_primes(limit)}")
Prime numbers up to 7: [2, 3, 5, 7]
```

The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help), a search bar, and a status bar at the bottom indicating "Connected to Python 3 Google Compute Engine backend".

# References & Credits

Python Documentation - <https://docs.python.org/3/>

Number Theory Concepts -  
<https://mathworld.wolfram.com/PrimeNumber.html>

Google Colab Guide - <https://colab.research.google.com/>

Developed by: [Your Name]

Institution: [Your Institution]

This project was developed as part of a study on prime number detection and generation using Python.