Name – **Arjun Singh Pundir**

Roll no- **202401100400047**

Branch – **CSE (AI & ML)**

Project – Prime Number generator and checker

Introductio:

1. Introduction

Prime numbers are natural numbers greater than 1 that have only two factors: 1 and themselves. The goal of this project is to develop a Python program that can:

1. Check whether a given number is prime
2. Generate all prime numbers up to a given limit

This project is implemented using Python in Google Colab, utilizing efficient algorithms for prime checking and generation. Below is an overview of our approach

2. Methodology

Prime Checking

- We use the basic divisibility rule:
- A number n is prime if it is not divisible by any number from 2 to √n.
- We optimize the function by skipping even numbers after checking 2.

Prime Generation

- The program generates primes up to a given number N using:
- Brute force method (checking divisibility for each number).
- Sieve of Eratosthenes for efficiency in larger cases.

Implementation

- The code runs in Google Colab with interactive user inputs.
- Results are displayed clearly for both checking and generation..

3. Code

```
# Import the required library for visualization
import matplotlib.pyplot as plt


def is_prime(n):
    """
    Function to check if a number is prime.
    A number is prime if it is greater than 1 and not divisible by
    any number from 2 to the square root of n.
```

```
    """
    if n < 2:  # Prime numbers start from 2
        return False
    for i in range(2, int(n**0.5) + 1):  # Loop through potential divisors
        if n % i == 0:  # If n is divisible by any of these, it's not prime
            return False
    return True  # Return True if no divisors were found

# Take user input and convert it to an integer
num = int(input("Enter a number to check if it's prime: "))

# Check if the number is prime and store the result
if is_prime(num):
    print(f"{num} is a prime number.")  # Display message for prime numbers
    bars = [num]  # Single bar for prime numbers
    heights = [1]  # Set height as 1 for prime numbers
else:
    print(f"{num} is not a prime number.")  # Display message for non-prime numbers
    bars = [num, num+1]  # Two bars for better visualization
    heights = [1, 2]  # Different heights to indicate non-prime status

# Create a bar chart for visualization
```

```
plt.figure(figsize=(5, 3))  # Set figure size
plt.bar(bars, heights, color='green' if is_prime(num) else 'red')  # Green for prime, red for non-prime
plt.xlabel("Number")  # Label for x-axis
plt.ylabel("Prime Status")  # Label for y-axis
plt.title(f"Visualization of {num}")  # Set title based on input number
plt.show()  # Display the plot
```

```
# Import the necessary library for visualization
import matplotlib.pyplot as plt

def is_prime(n):
    """
    Function to check if a number is prime.
    A number is prime if it is greater than 1 and not divisible
    by any number from 2 to sqrt(n).
    """
    if n < 2:  # Prime numbers start from 2
        return False
    for i in range(2, int(n**0.5) + 1):  # Loop through possible divisors
        if n % i == 0:  # If divisible, it's not a prime number
            return False
```

```
    return True  # Return True if no divisors were found

def generate_primes(limit):
    """
    Function to generate prime numbers up to a given limit.
    It iterates through numbers from 2 to the limit and checks if they are prime.
    """
    primes = []  # Initialize an empty list to store prime numbers
    for num in range(2, limit + 1):  # Iterate through numbers from 2 to limit
        if is_prime(num):  # Check if the number is prime using is_prime function
            primes.append(num)  # Add the prime number to the list
    return primes  # Return the list of prime numbers

# Take user input and convert it to an integer
limit = int(input("Enter a limit to generate prime numbers up to: "))

# Generate the list of prime numbers
primes = generate_primes(limit)

# Print the prime numbers found
print(f"Prime numbers up to {limit}: {primes}")
```

 # Visualization: Plot the prime numbers as a bar chart

plt.figure(figsize=(10, 5)) **# Set the figure size**

plt.bar(primes, primes, color='cyan') **# Create bars at prime number positions**

plt.xlabel("Prime Numbers") **# Label for x-axis**

plt.ylabel("Value") # Label for y-axis

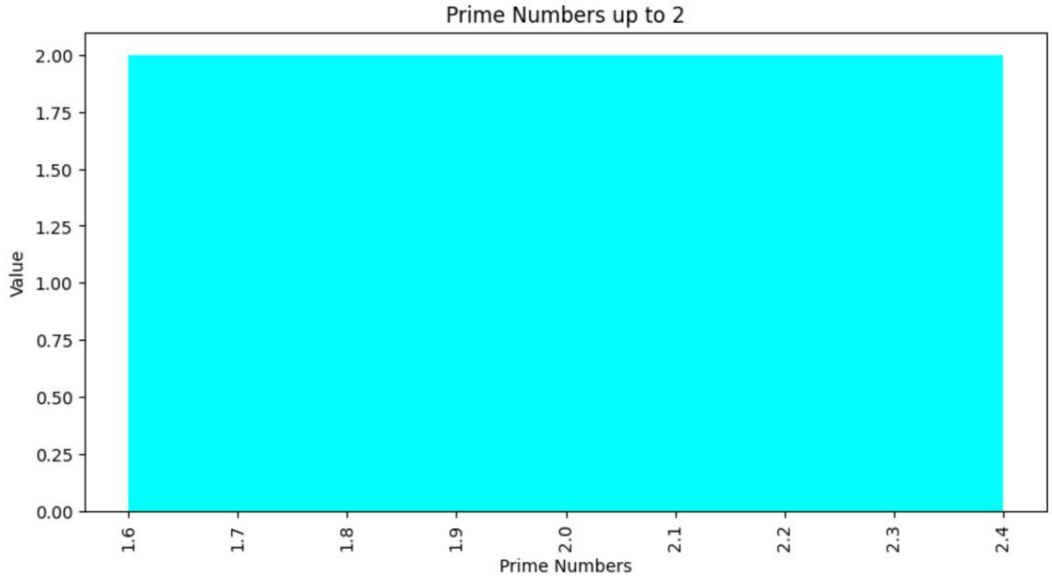
plt.title(f"Prime Numbers up to {limit}") **# Set title dynamically**

plt.xticks(rotation=90) **# Rotate x-axis labels for better visibility**

plt.show() **# Display the plot**

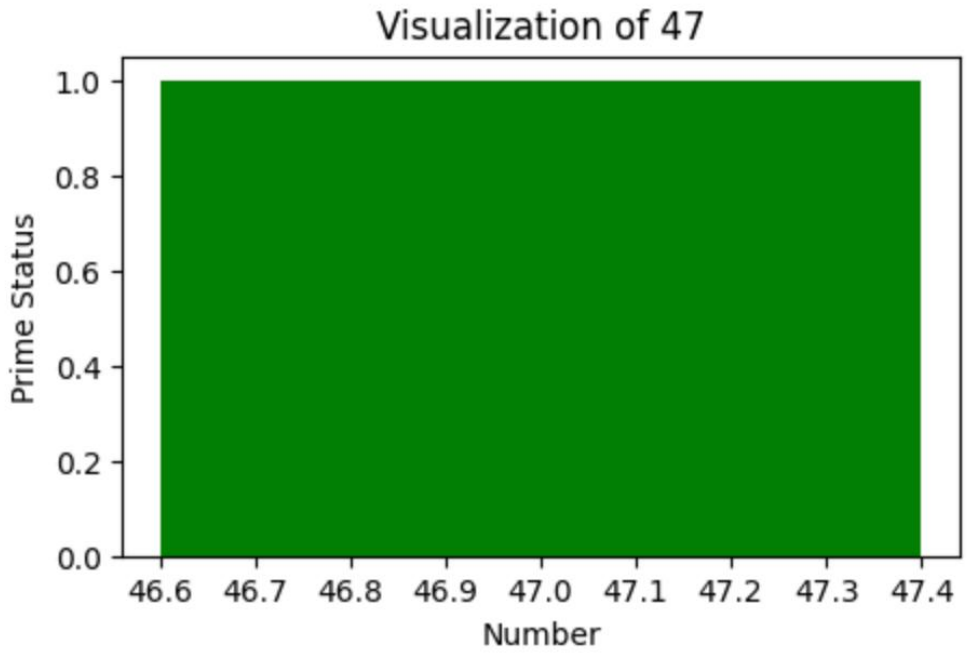
4. Output

```
Enter a limit to generate prime numbers up to: 2
Prime numbers up to 2: [2]
```



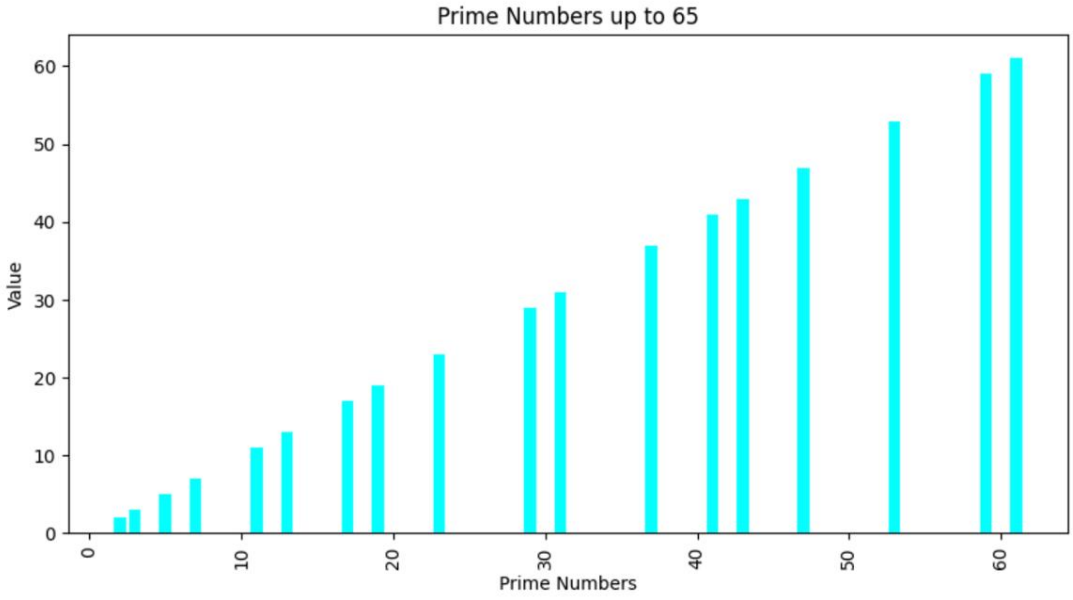
Output till prime number unto 2

```
Enter a number to check if it's prime: 47
47 is a prime number.
```



Output showing 47 is prime number

```
Enter a limit to generate prime numbers up to: 65
Prime numbers up to 65: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61]
```



Output showing all prime number till 47.

Special credit to Bikki sir