

KIET Group of Institutions, Ghaziabad

Course: Introduction to AI

Project Title: Prime Number Generator and Checker

Course: Introduction to AI

Submitted by: Mohd Suleman Aqdam

Roll Number: [202401100300156]

Date: [11/03/2025]

2. Introduction

Objective

The objective of this project is to create a Python program that can:

- Check whether a given number is prime.
- Generate a list of the first n prime numbers.

Scope

- The program checks for prime numbers using an efficient trial division method.
- It generates prime numbers using a loop-based approach.
- The program provides a user-friendly interface for interaction.

Definition of Prime Number

A prime number is a natural number greater than 1 that has exactly two divisors: 1 and itself. Examples include 2, 3, 5, 7, 11, etc.

3. Methodology

Step 1: Prime Checking

- A function `is_prime(num)` is created to check if a given number is prime.
- The function checks divisibility from 2 to \sqrt{n} for optimization.

Step 2: Prime Number Generation

- A function `generate_primes(n)` is created to generate the first `n` prime numbers.
- It starts checking from 2 and continues until the desired number of prime numbers is found.

Step 3: User Interaction

- A `main()` function is created to handle user input and display results.
 - The program gives two options:
 1. Check if a number is prime.
 2. Generate a list of prime numbers.
-

4. Code Typed

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

# Function to generate the first 'n' prime numbers
def generate_primes(n):
    primes = []
    num = 2
    while len(primes) < n:
        if is_prime(num):
            primes.append(num)
        num += 1
    return primes

# Main function to handle user interaction
def main():
    print("1. Check if a number is prime")
    print("2. Generate a list of prime numbers")

    choice = input("Enter your choice (1/2): ")

    if choice == '1':
        num = int(input("Enter a number to check: "))
        if is_prime(num):
            print(f"{num} is a prime number.")
        else:
            print(f"{num} is not a prime number.")

    elif choice == '2':
        n = int(input("How many prime numbers do you want to generate? "))
        primes = generate_primes(n)
        print(f"The first {n} prime numbers are: {primes}")
```

```
        else:
            print("Invalid choice!")

if __name__ == "__main__":
    main()
```

5. Screenshots

```
1. Check if a number is prime
2. Generate a list of prime numbers
3. Exit
Enter your choice (1/2/3): 1
Enter a number to check: 56
56 is not a prime number.
Do you want to continue? (yes/no): yes

1. Check if a number is prime
2. Generate a list of prime numbers
3. Exit
Enter your choice (1/2/3): 1
Enter a number to check: 3
3 is a prime number.
Do you want to continue? (yes/no): yes

1. Check if a number is prime
2. Generate a list of prime numbers
3. Exit
Enter your choice (1/2/3): 2
How many prime numbers do you want to generate? 7
The first 7 prime numbers are: [2, 3, 5, 7, 11, 13, 17]
```

```
1. Check if a number is prime
2. Generate a list of prime numbers
3. Exit
Enter your choice (1/2/3): no
Invalid choice! Please try again.

1. Check if a number is prime
2. Generate a list of prime numbers
3. Exit
Enter your choice (1/2/3): 2
How many prime numbers do you want to generate? 10
The first 10 prime numbers are: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]

1. Check if a number is prime
2. Generate a list of prime numbers
3. Exit
Enter your choice (1/2/3): 3
Exiting the program. Goodbye!
```

6. Conclusion

The project successfully implemented a Python program to:

- ✓ ☐ Check whether a number is prime.
- ✓ ☐ Generate the first n prime numbers efficiently.
- ✓ ☐ The code is optimized using trial division up to \sqrt{n} , ensuring fast execution.