Lab : Prime Number Checker – Documentation

1. Introduction

A prime number is a natural number greater than 1 that has no divisors other than 1 and itself. This program is designed to take an integer as input and determine whether it is a prime number or not. The program has been developed in Python using the Thonny IDE, which is a beginner-friendly environment for learning and developing Python programs.

---

2. Objectives

- To create a program that checks if a given number is prime.
- To handle incorrect or invalid inputs gracefully.
- To provide clear user-friendly output.

---

3. Tools and Technologies

- **Programming Language:** Python 3
- **IDE:** Thonny
- **Libraries Used:** `math` (for square root calculation)

---

4. Program Logic

The prime checking algorithm works as follows:

1. If the number is less than or equal to 1, it is not prime.
2. If the number is 2 or 3, it is prime.
3. If the number is divisible by 2, it is not prime.
4. Otherwise, check only odd numbers from 3 up to the square root of the given number.
5. If no divisor is found, the number is prime.

---

5. Flow of Execution

1. **User Input:**
    - The user is prompted to enter a whole number.
    - Input is validated to ensure it is an integer (no decimals or letters allowed).
2. **Prime Check Function (`is_prime`):**

    - Receives the integer as a parameter.
    - Implements the prime number checking logic described above.
3. **Output:**
    - Displays whether the number is prime or not.
    - Displays a special message for numbers ≤ 1.

---

6. Sample Output

Case 1: Prime number

```
Enter a whole number: 17
17 is a prime number.
```

Case 2: Non-prime number

```
Enter a whole number: 18
18 is not a prime number.
```

Case 3: Invalid input

```
Enter a whole number: 3.5
Please enter a whole number (integer), not a decimal.
```

---

6. Advantages of This Program

- Easy to use and beginner-friendly.
- Efficiently checks primality by reducing unnecessary calculations.
- Handles invalid input with clear error messages.

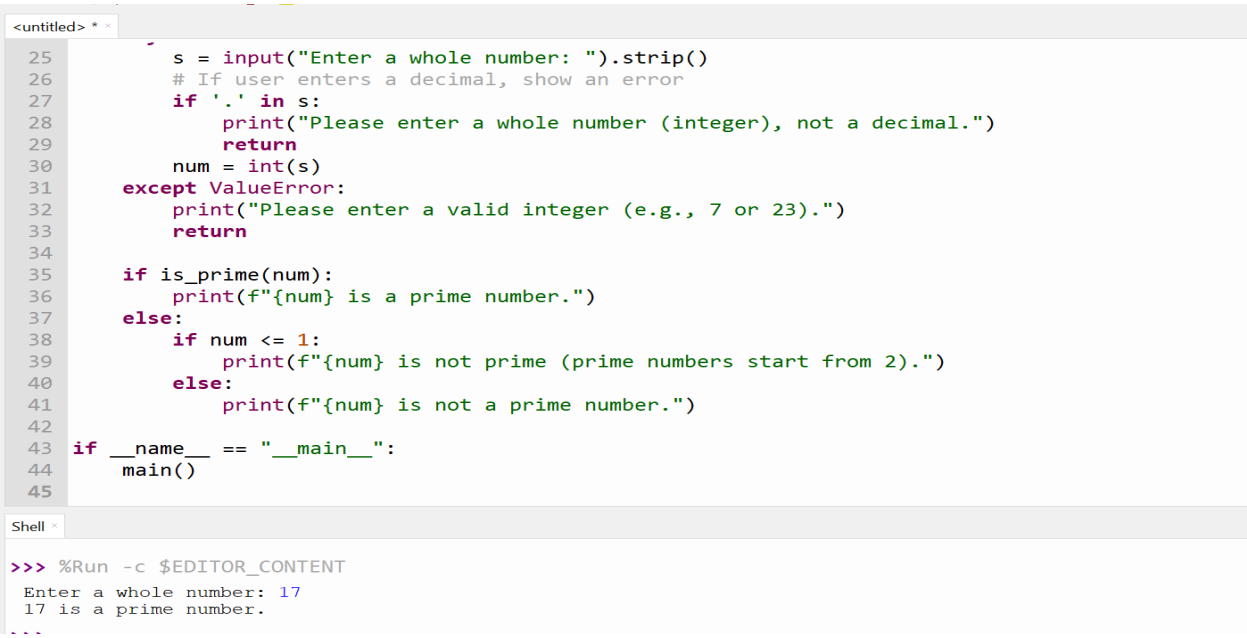
---

8. Possible Enhancements

- Allow the user to check multiple numbers at once.
- Implement faster algorithms for very large numbers (e.g., Miller–Rabin test).
- Add a graphical user interface (GUI) for better user interaction.

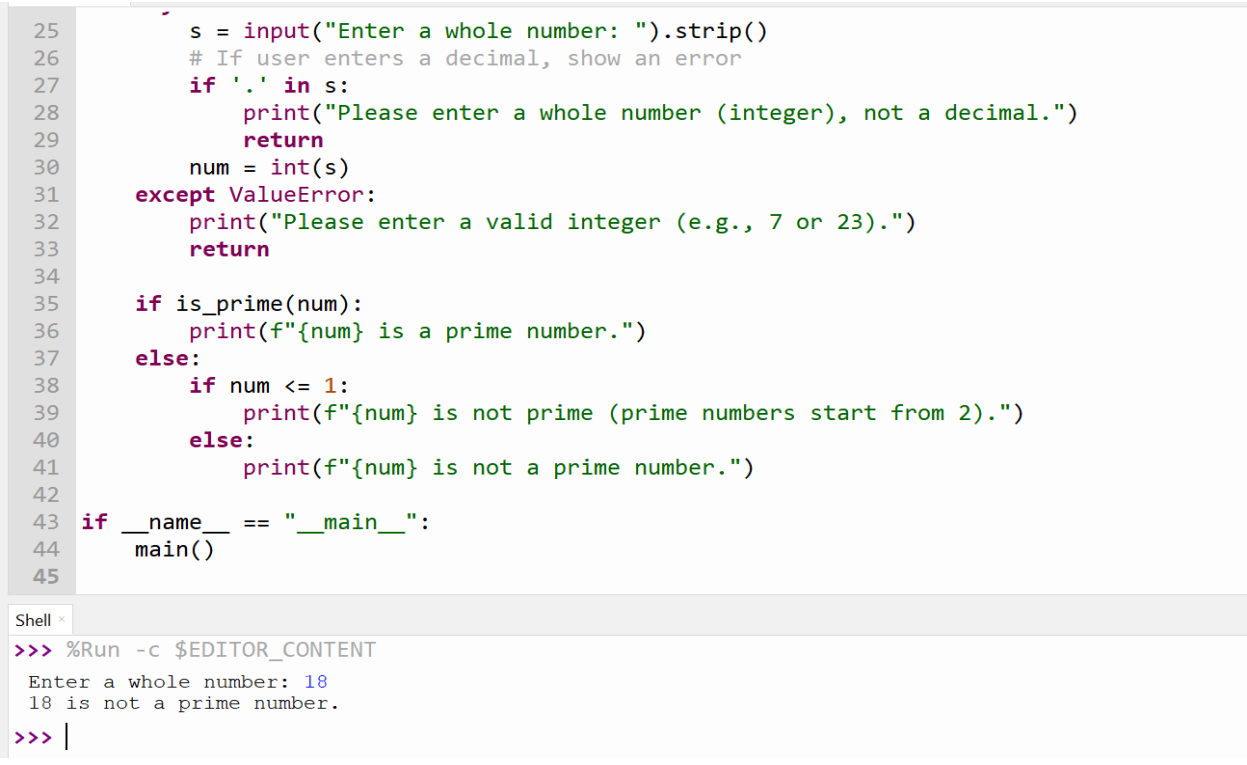
9. Results Screenshots:

- **Case 1:**



```
<untitled> *
25          s = input("Enter a whole number: ").strip()
26          # If user enters a decimal, show an error
27          if '.' in s:
28              print("Please enter a whole number (integer), not a decimal.")
29              return
30          num = int(s)
31      except ValueError:
32          print("Please enter a valid integer (e.g., 7 or 23).")
33          return
34
35      if is_prime(num):
36          print(f"{num} is a prime number.")
37      else:
38          if num <= 1:
39              print(f"{num} is not prime (prime numbers start from 2).")
40          else:
41              print(f"{num} is not a prime number.")
42
43  if __name__ == "__main__":
44      main()
45
Shell
>>> %Run -c $EDITOR_CONTENT
 Enter a whole number: 17
 17 is a prime number.
```

- **Case 2:**



```
25          s = input("Enter a whole number: ").strip()
26          # If user enters a decimal, show an error
27          if '.' in s:
28              print("Please enter a whole number (integer), not a decimal.")
29              return
30          num = int(s)
31      except ValueError:
32          print("Please enter a valid integer (e.g., 7 or 23).")
33          return
34
35      if is_prime(num):
36          print(f"{num} is a prime number.")
37      else:
38          if num <= 1:
39              print(f"{num} is not prime (prime numbers start from 2).")
40          else:
41              print(f"{num} is not a prime number.")
42
43  if __name__ == "__main__":
44      main()
45
Shell
>>> %Run -c $EDITOR_CONTENT
 Enter a whole number: 18
 18 is not a prime number.
>>>
```

- **Case 3:**

```
    if is_prime(num):
        print(f"{num} is a prime number.")
    else:
        if num <= 1:
            print(f"{num} is not prime (prime numbers start from 2).
        else:
            print(f"{num} is not a prime number.")

if __name__ == "__main__":
    main()

```

Shell

```
>>> %Run -c $EDITOR_CONTENT
 Enter a whole number: 3.5
 Please enter a whole number (integer), not a decimal.
>>>
```

The End