1. Cover Page

Project Title: Series of Digit Program
Course: Introduction To Problem Solving and Programming – CSE(1021)
Submitted by: Pari Bansal (25BCE11060)

Date: [24/11/2025]

2. Introduction

This project demonstrates string manipulation and sequence logic in Python. The focus is on generating and analysing concatenated number series, utilizing user input to extract digit positions.

3. Problem Statement

Automate the generation of a number series by concatenating sequential integers, and allow users to query the digit at a specified position. Additionally, provide logic to examine any given series and report the last digit and its exact position.

4. Functional Requirements

- Generate a concatenated number series up to a user - specified position.
- Output the digit at any position in the series.
- Accept a pre - defined series and return the last digit and its position.
- Provide user - friendly prompts and clear results.

5. Non - functional Requirements

- Code must be written in Python 3 and be readable and well - commented.
- Input validation and error handling for user entries.
- Efficient execution for moderate input sizes.
- Clear output formatting.

6. System Architecture

- **Input Layer:** Receives user input for target position or custom series.
- **Processing Layer:** Builds the sequence or analyses the given string.
- **Output Layer:** Prints the required digit and its position.

7. Design Diagrams

- **Use Case Diagram:** Illustrates user interactions (User inputs position; system outputs digit).
- **Workflow Diagram:** Sequence → User input → Process → Output.
- **Sequence Diagram:** Initiation, function call, sequence generation, digit lookup, result display.
- **Class/Component Diagram:** Functions for series generation and position finding.

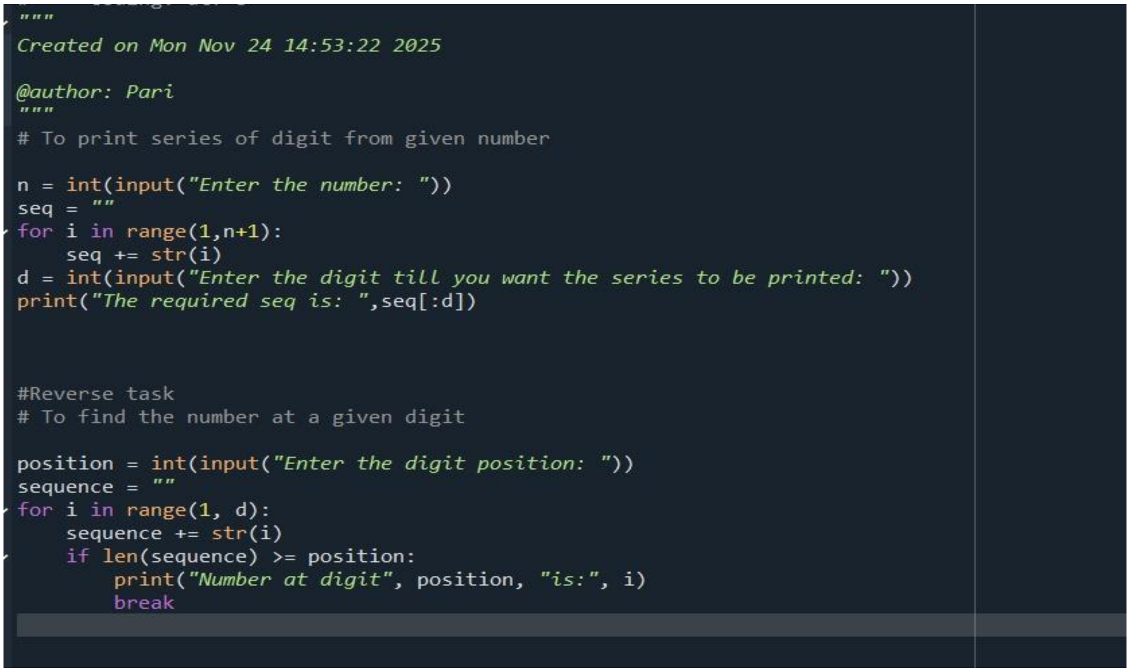
8. Design Decisions & Rationale

- Simple procedural approach for clarity.
- Separation of concerns: One function for generation, another for analysis.
- Print statements for instant feedback.

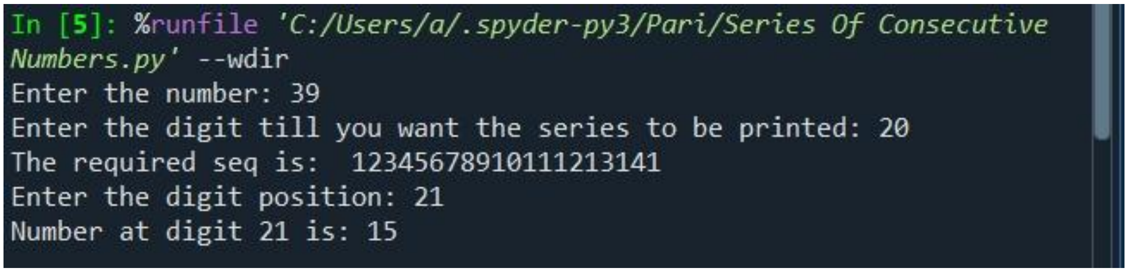
9. Implementation Details

- Python functions print_number_series(n) and find_end_position_of_number_series(series).
- Loop constructs and string concatenation.
- Inline comments for step-wise explanation.

10. Screenshots / Results



```
"""
Created on Mon Nov 24 14:53:22 2025

@author: Pari
"""
# To print series of digit from given number

n = int(input("Enter the number: "))
seq = ""
for i in range(1,n+1):
    seq += str(i)
d = int(input("Enter the digit till you want the series to be printed: "))
print("The required seq is: ",seq[:d])



#Reverse task
# To find the number at a given digit

position = int(input("Enter the digit position: "))
sequence = ""
for i in range(1, d):
    sequence += str(i)
    if len(sequence) >= position:
        print("Number at digit", position, "is:", i)
        break
```



```
In [5]: %runfile 'C:/Users/a/.spyder-py3/Pari/Series Of Consecutive
Numbers.py' --wdir
Enter the number: 39
Enter the digit till you want the series to be printed: 20
The required seq is:  12345678910111213141
Enter the digit position: 21
Number at digit 21 is: 15
```

11. Testing Approach

- Multiple test cases for various position values and input sequences.
- Boundary tests for low and high values.
- Validation of output at different positions.

12. Challenges Faced

- Managing string length efficiently.
- Handling user input errors.
- Ensuring clarity in output statements.

13. Learnings & Key Takeaways

- Strengthened understanding of string manipulation and indexing in Python.
- Practiced user input validation and clear output formatting.
- Gained experience in structuring small-scale Python projects.

14. Future Enhancements

- Expand to support larger series efficiently.
- Integrate graphical user interface for input and output.
- Store series and positions in a database for audit/tracking.

15. References

- Python official documentation
- Course lecture notes
- Similar template resources online