Design Document - Recursive Digit Addition

Problem Statement

The goal of this program is to take two numbers from the user, add them together, and repeatedly separate the least significant digit from the result and add it to the remaining digits. This process continues until a single-digit result is obtained.

Example

Input: 87300, 45

Summed Number: 87345

Output: 9

Design Overview

The problem can be solved using a recursive algorithm. The main idea is to perform the following steps:

1. Take two numbers from the user.
2. Add the two numbers together.
3. If the result is a single digit, return it.
4. If the result is not a single digit, separate the least significant digit (rightmost digit) and add it to the remaining digits.
5. Recursively repeat the process with the updated result until a single-digit result is obtained.

Pseudocode

```
pythonCopy code
```

```
function recursive_digit_addition(number): if number is a single digit: return numberelse: sum = 0 while number > 0:
sum += number % 10 # Get the least significant digit and add it to the sum number = number // 10 # Remove the least
significant digit returnrecursive_digit_addition(sum) # Main program input_number_1 = get_user_input("Enter the first
number: ") input_number_2 = get_user_input("Enter the second number: ") result = input_number_1 + input_number_2
final_result = recursive_digit_addition(result)print("The final result is:", final_result)
```

Functionality Breakdown

1. `recursive_digit_addition(number)` : This is a recursive function that takes an integer `number` as input. It checks if the number is a single digit. If it is, it returns the number; otherwise, it separates the least significant digit, adds it to the remaining digits, and recursively calls itself with the updated sum.
2. Main Program:
   - Get user input for two numbers (`input_number_1` and `input_number_2`).
   - Calculate the initial sum of the two numbers.

- Call the `recursive_digit_addition` function with the initial sum.
- Print the final result.

Input and Output

The program will take two integers as input and provide a single-digit result as output.

Error Handling

- The program should handle cases where the user enters non-integer input or invalid input gracefully, providing appropriate error messages and requesting valid input.

Testing

Test the program with various input values to ensure it works correctly and handles different scenarios. Test cases should include both single-digit and multi-digit input numbers.

Conclusion

This design document outlines the approach and pseudocode for a recursive digit addition program that takes two numbers from the user, adds them together, and recursively separates and adds digits until a single-digit result is obtained. The implementation should follow the pseudocode and include error handling for invalid input.

** This document was generated by ChatGPT and edited by myself.**