LAB01.1

FIT-HCMUS

Exercise 1

Write a function `sum_of_odd_digits(n)` that takes a positive integer n as input and returns the sum of all odd digits in n.

Exercise 2

Write a function `reverse_number(n)` that takes a positive integer n and returns the number n with its digits reversed.

Exercise 3

Write a function `nearest_square(n)` that takes a positive integer n and finds the nearest square number less than or equal to n. A square number is a number of the form x^2, where x is an integer.

Exercise 4

Write a function `is_palindrome(n)` that checks whether a given positive integer n is a palindrome. A palindrome is a number that reads the same backward as forward.

Exercise 5

Write a function `digit_power_sum(n)` that takes a positive integer n and calculates the sum of its digits, each raised to the power of its respective position (counted from right to left, starting at 1).

Ex

```
sum_of_digit_squares(123)  # Output: 1^2 + 2^2 + 3^2 = 1 + 4 + 9 = 14
```

Exercise 6

Write a function `sum_max_min_digits(n)` that takes a positive integer n and returns the sum of the largest and smallest digit in n.

Exercise 7

Write a function `digit_count(n)` that takes a positive integer n and returns the number of digits in n that are divisible by 3.

Exercise 8

Write a function `split_and_compare(n)` that takes a positive integer n and splits its digits into two groups: one with digits at even positions and the other with digits at odd positions (positions counted from right to left, starting from 1). Return the absolute difference between the sum of digits at even positions and the sum of digits at odd positions.

Exercise 9

Write a function `max_digit_subtraction_sequence(n)` that takes a positive integer n and performs a series of operations where, at each step, you subtract the largest digit of n from n. The process should continue until n becomes zero or a negative number. Return the number of operations required to reduce n to zero or below.

Ex:

```
max_digit_subtraction_sequence(32)  # Output: 6
# Explanation:
# Step 1: n = 32 -> Subtract 3 -> 32 - 3 = 29
# Step 2: n = 29 -> Subtract 9 -> 29 - 9 = 20
# Step 3: n = 20 -> Subtract 2 -> 20 - 2 = 18
# Step 4: n = 18 -> Subtract 8 -> 18 - 8 = 10
# Step 5: n = 10 -> Subtract 1 -> 10 - 1 = 9
# Step 6: n = 9 -> Subtract 9 -> 9 - 9 = 0 (Total of 6 operations)
```