AI ASSISTED CODING 2303A52490

ASSIGNMENT 4

Task 1: Zero-shot Prime Number Checker

prompt : Write a Python function to determine whether a given number is prime

```
def is_prime(n):
    if n <= 1:
        return False
    if n <= 3:
        return True
    if n % 2 == 0 or n % 3 == 0:
        return False
    i = 5
    while i * i <= n:
        if n % i == 0 or n % (i + 2) == 0:
            return False
        i += 6
    return True

test_numbers = [2, 3, 4, 17, 25, 97, 100]
for num in test_numbers:
    print(f"{num} is prime: {is_prime(num)}")
```

output

```
2 is prime: True
3 is prime: True
4 is prime: False
17 is prime: True
25 is prime: False
97 is prime: True
100 is prime: False
```

Task 2: One-shot List Sum Function

prompt : Write a Python function that calculates the sum of elements in a list

```
def sum_list(numbers):
    total = 0
    for num in numbers:
        total += num
    return total

print(sum_list([1, 2, 3, 4]))
print(sum_list([10, 20, 30]))
print(sum_list([-1, 0, 1]))
```

output :

```
10
60
0
```

Task 3: Few-shot Digit Extraction Function

prompt : Write a Python function that extracts digits from an alphanumeric string

```python
def extract_digits(input_string):
    digits = ""
    for char in input_string:
        if char.isdigit():
            digits += char
    return digits

example1 = "abc123xyz"
example2 = "Phone: 987-654-3210"
example3 = "ID# A4B5C6"

print(f"Input: '{example1}' → Output: '{extract_digits(example1)}'")
print(f"Input: '{example2}' → Output: '{extract_digits(example2)}'")
print(f"Input: '{example3}' → Output: '{extract_digits(example3)}'")
```

output

```
Input: 'abc123xyz' → Output: '123'
Input: 'Phone: 987-654-3210' → Output: '9876543210'
Input: 'ID# A4B5C6' → Output: '456'
```

Task 4: Zero-shot vs Few-shot Vowel Counter

prompt: Write a Python function that counts the number of vowels in a string

Zero-shot Version

```python
def count_vowels_zero_shot(text):
    vowels = "aeiouAEIOU"
    count = 0
    for char in text:
        if char in vowels:
            count += 1
    return count

print("Zero-shot version:")
print(count_vowels_zero_shot("hello"))
print(count_vowels_zero_shot("PYTHON"))
print(count_vowels_zero_shot("rhythm"))
```

output

```
Zero-shot version:
2
1
0
```

Few-shot Version

```python
def count_vowels_few_shot(text):
    vowel_count = 0
    text = text.lower()
    for char in text:
        if char in ['a', 'e', 'i', 'o', 'u']:
            vowel_count += 1
    return vowel_count

print("Few-shot version:")
print(f"Input: 'apple' → Output: {count_vowels_few_shot('apple')}")
print(f"Input: 'EDUCATION' → Output: {count_vowels_few_shot('EDUCATION')}")
print(f"Input: 'sky' → Output: {count_vowels_few_shot('sky')}")
```

output

```
Few-shot version:
Input: 'apple' → Output: 2
Input: 'EDUCATION' → Output: 5
Input: 'sky' → Output: 0
```

Task 5: Few-shot Minimum of Three Numbers

prompt : Write a Python function that determines the minimum of three numbers without using the built-in min() function

```python
def min_of_three(a, b, c):
    if a <= b and a <= c:
        return a
    elif b <= a and b <= c:
        return b
    else:
        return c

print(f"min_of_three(5, 10, 3) = {min_of_three(5, 10, 3)}")
print(f"min_of_three(-1, -5, -3) = {min_of_three(-1, -5, -3)}")
print(f"min_of_three(7, 7, 2) = {min_of_three(7, 7, 2)}")
```

output

```
min_of_three(5, 10, 3) = 3
min_of_three(-1, -5, -3) = -5
min_of_three(7, 7, 2) = 2
```