School of Computer Science and Artificial Intelligence

Lab Assignment # 4

Name of Student : Gunturu Devi Prasad
Enrollment No.   : 2303A51538
Batch No.   : 22

Task Description-1
• Zero-shot: Prompt AI with only the instruction. Write a Python function to determine whether a given number is prime
Expected Output-1
• A basic Python function to check if a number is prime, demonstrating correct logical conditions without relying on examples or additional context

Prompt: write a function to check if a number is prime demonstrating correct logical conditions without relying on examples or additional context in user-defined range.

Code:-

```python
# Task1
# write a function to check if a number is prime demonstrating cor
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
# Example usage:
number = int(input("Enter a number to check if it's prime: "))
if is_prime(number):
    print(f"{number} is a prime number.")
else:
    print(f"{number} is not a prime number.")
```

Output:-

```
p\Ai Assistant coding\Assignment4.2.py'
Enter a number to check if it's prime: 8
8 is not a prime number.
```

Justification:-

The function checks for primality by testing divisibility from 2 up to the square root of the number, which is an efficient and standard method for prime checking.

Task Description-2
• One-shot: Provide one example: Input: [1, 2, 3, 4], Output: 10 to help AI generate a function that calculates the sum of elements in a list.
Expected Output-2
• A correct conversion function guided by the single example.

Prompt: write a program to calculate the sum of given array input:- [1, 2, 3, 4] output:- 10 in dynamic way.

Code:-

```
Assignments Codes >  Assignment4.py >  sum_of_array
19    # Task2
20    #write a program to calculate the sum of given array input:-
21    def sum_of_array(arr):
22        total = 0
23        for num in arr:
24            total += num
25        return total
26    # Example usage:
27    array_input = input("Enter numbers separated by spaces: ")
28    array = list(map(int, array_input.split()))
29    result = sum_of_array(array)
30    print(f"The sum of the array is: {result}")
31
```

Output:-

```
Enter numbers separated by spaces: 7 8 5 4 1 9 78
The sum of the array is: 112
```

Justification:-

The program takes dynamic input from the user, allowing for any array of integers to be summed, rather than being limited to a predefined array.

Task Description-3
• Few-shot: Give 2–3 examples to create a function that extracts digits from an alphanumeric string.
Expected Output-3
• Accurate function that returns only the digits from alphanumeric string.

Prompt:write a program to alphanumeric string and accurate function only digits from alphanumeric string.

Code:-

```
Assignments Codes > 🐍 Assignment4.py > ...
33    # Task3
34    # write a program to alphanumeric string and accurate function only digits from alphanumeric string.
35    def extract_digits(alphanumeric_str):
36        digits = ''.join(filter(str.isdigit, alphanumeric_str))
37        return digits
38    # Example usage:
39    alphanumeric_input = input("Enter an alphanumeric string: ")
40    digits_only = extract_digits(alphanumeric_input)
41    print(f"The digits extracted from the string are: {digits_only}")
42
```

Output:-

```
Enter an alphanumeric string: deviprasad15380
The digits extracted from the string are: 15380
```

Justification:-

The function uses the built-in filter function along with str.isdigit to efficiently extract only the digit characters from the input string. This approach is concise and leverages Python's capabilities for string manipulation.

Task Description-4

• **Compare zero-shot vs few-shot prompting for generating a function that counts the number of vowels in a string.**

Expected Output-4

• **Output comparison + student explanation on how examples helped the model.**

Prompt:write a program to function that counts the number of vowels in a given string.

Code:-

```
Assignments Codes > 🐍 Assignment4.py > ...
44    # Task4
45    # write a program to function that counts the number of vowels in a given string.
46    def count_vowels(input_str):
47        vowels = 'aeiouAEIOU'
48        count = sum(1 for char in input_str if char in vowels)
49        return count
50    # Example usage:
51    string_input = input("Enter a string: ")
52    vowel_count = count_vowels(string_input)
53    print(f"The number of vowels in the string is: {vowel_count}")
54
```

Output:-

```
Enter a string: Gunturu Devi Prasad cse sru student
The number of vowels in the string is: 11
```

Justification:-
The function iterates through each character in the input string and checks if it is a vowel by seeing if it exists in the predefined string of vowels. It uses a generator expression to count the occurrences efficiently.

Task Description-5
• Use few-shot prompting with 3 sample inputs to generate a function that determines the minimum of three numbers without using the built-in min() function.

Expected Output-5
• A function that handles all cases with correct logic based on example patterns.

Prompt:write a program to generate function that handles the minimum of three numbers withput using builtin min() function.

Code:-

```python
Assignments Codes > Assignment4.py > ...
55
56    # Task5
57    # write a program to generate function that handles the minimum o
58    def minimum_of_three(a, b, c):
59        if a <= b and a <= c:
60            return a
61        elif b <= a and b <= c:
62            return b
63        else:
64            return c
65    # Example usage:
66    num1 = float(input("Enter first number: "))
67    num2 = float(input("Enter second number: "))
68    num3 = float(input("Enter third number: "))
69    min_number = minimum_of_three(num1, num2, num3)
70    print(f"The minimum of the three numbers is: {min_number}")
71
```

Output:-

```
Enter first number: 25
Enter second number: 69
Enter third number: 88
The minimum of the three numbers is: 25.0
```

Justification:-

The provided code snippets effectively address each of the specified tasks using clear and logical implementations. Each function is designed to perform its intended operation without relying on external libraries or built-in functions that would simplify the task, thus demonstrating a solid understanding of fundamental programming concepts.