

School of Computer Science and Artificial Intelligence

Lab Assignment # 4

Name of Student : K.SAIKIRAN
Enrollment No. : 2303A51539
Batch No. : 22

Task-1:

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 python code function to determine whether the number is prime or not

Code:-

```
#write a python function whether a given number is prime or not
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 = input("Enter a number:")
number = int(number)
result = is_prime(number)
print(f'Is the number {number} prime? {result}')
```

Output:-

```
PS C:\Users\Dell\Desktop\PYTHON> & C:/Users/Dell/AppData/Local/Programs/Python/Python312/python.exe
Enter a number:12
Is the number 12 prime? False
```

Observation

- The function is correct but basic.
- Logic is simple without optimization.

Task Description – 2 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 python program to generate a function that calculates the sum of elements in a list.

Code:-

```
#write a python program to generate a function that calculates the sum of elements in a list
def sum_of_elements(elements):
    return sum(elements)
#Example usage:
input_list = list(map(int, input("Enter numbers separated by commas: ").split(',')))
total = sum_of_elements(input_list)
print(f'The sum of elements in the list {input_list} is {total}')
```

Output:

```
PS C:\Users\DeII\Desktop\PYTHON> & C:/Users/DeII/AppData/Local/Programs/Python/Python312/python.exe
Enter numbers separated by commas: 1,2,3,4,5
The sum of elements in the list [1, 2, 3, 4, 5] is 15
```

Observation

- The example helps AI understand expected behaviour.
- Output accuracy improves compared to zero-shot.

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 python program to create a function that extracts digits from an alphanumeric string.

Code:

```
#write a python program to create a function that extracts digits from an alphanumeric string.
def extract_digits(alphanumeric_string):
    return ''.join(filter(str.isdigit, alphanumeric_string))
#Example usage:
input_string = input("Enter an alphanumeric string: ")
digits = extract_digits(input_string)
print(f'The digits extracted from the string "{input_string}" are: {digits}')
```

Output:

```
Enter an alphanumeric string: sai18kiran
The digits extracted from the string "sai18kiran" are: 18
PS C:\Users\Dell\Desktop\PYTHON>
```

Observation

- Code is accurate and reliable.
- Multiple examples reduce ambiguity.

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-3**
- Output comparison + student explanation on how examples helped the model.

Prompt:

Write a python program for generating a function that counts the number of vowels in a given string.

Code:

```
#write a python program for generating a function that counts the number of vowels in a given string.
def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    count = sum(1 for char in input_string if char in vowels)
    return count
#Example usage:
input_str = input("Enter a string: ")
vowel_count = count_vowels(input_str)
print(f'The number of vowels in the string "{input_str}" is: {vowel_count}')
```

Output:

```
Enter a string: saikiran
The number of vowels in the string "saikiran" is: 4
PS C:\Users\Dell\Desktop\PYTHON>
```

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 python code Minimum of three numbers by the in user input format

code:

```
#write a python program with 3 sample inputs to generate a function that determines the minimum of three numbers without
# using the built-in min() function.

def find_minimum(a, b, c):
    if a <= b and a <= c:
        return a
    elif b <= a and b <= c:
        return b
    else:
        return c

#Example usage:
input_numbers = tuple(map(int, input("Enter three numbers separated by commas: ").split(',')))
minimum_value = find_minimum(*input_numbers)
print(f"The minimum of the numbers {input_numbers} is: {minimum_value}")
```

Output:

```
Enter three numbers separated by commas: 7,6,8
The minimum of the numbers (7, 6, 8) is: 6
PS C:\Users\Dell\Desktop\PYTHON> 
```