Assignment-4.2

Name : Saisharan

Ht.no :2303A51434 Bt.no: 21

Course :Ai Assistant Coding

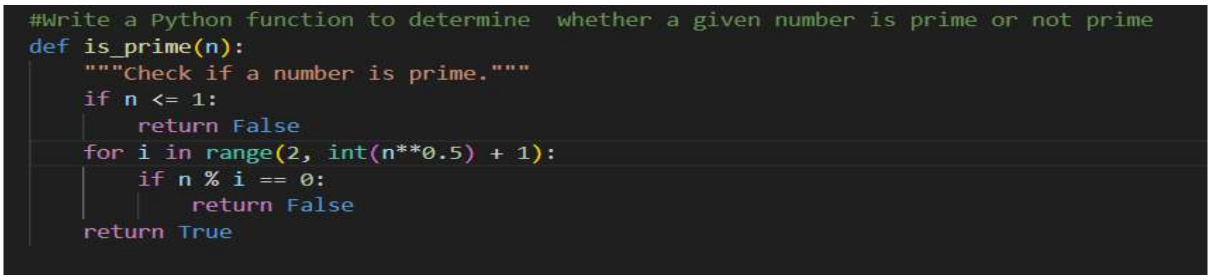
Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques

Task Description-1:

- Zero-shot: Prompt AI with only the instruction. Write a Python function to determine

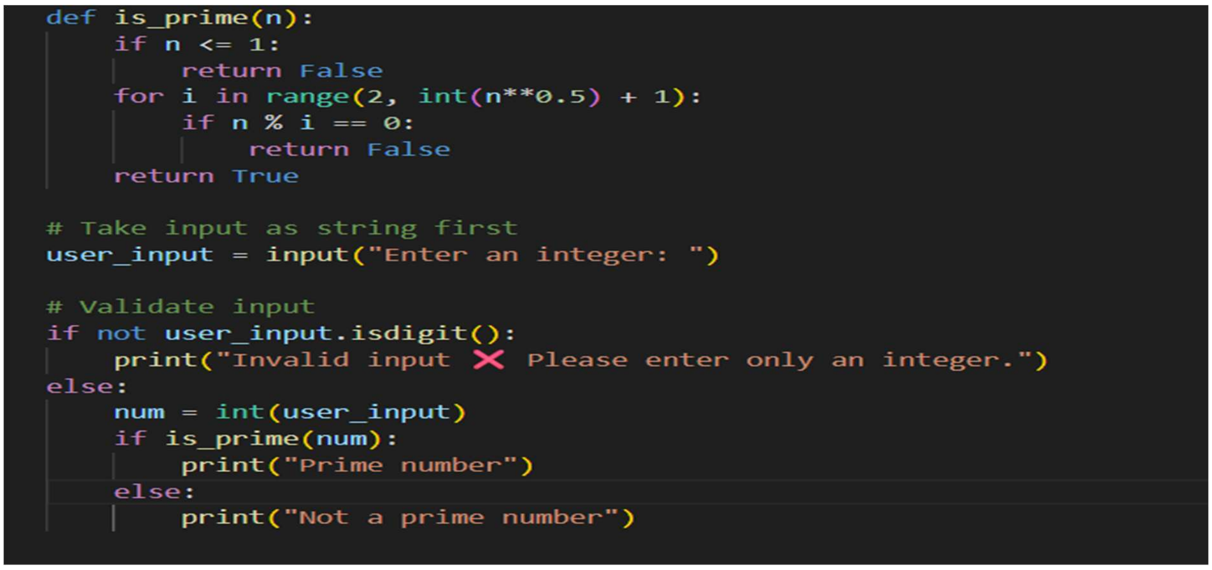
whether a given number is prime

Code:



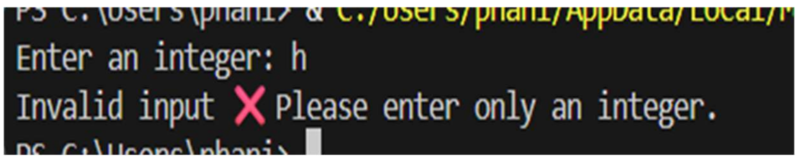
```python
#Write a Python function to determine  whether a given number is prime or not prime
def is_prime(n):
    """Check if a number is prime."""
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
```

When input is taken other than integer like string or special symbol

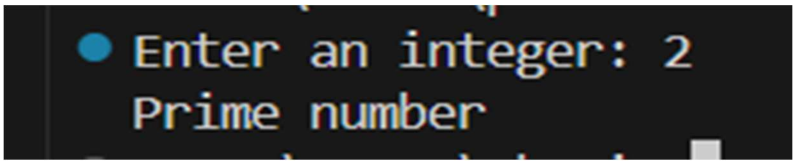


```python
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

# Take input as string first
user_input = input("Enter an integer: ")

# Validate input
if not user_input.isdigit():
    print("Invalid input ❌ Please enter only an integer.")
else:
    num = int(user_input)
    if is_prime(num):
        print("Prime number")
    else:
        print("Not a prime number")
```

Output:



```
PS C:\Users\phani> & C:/Users/phani/AppData/Local/
Enter an integer: h
Invalid input ❌ Please enter only an integer.
PS C:\Users\phani>
```

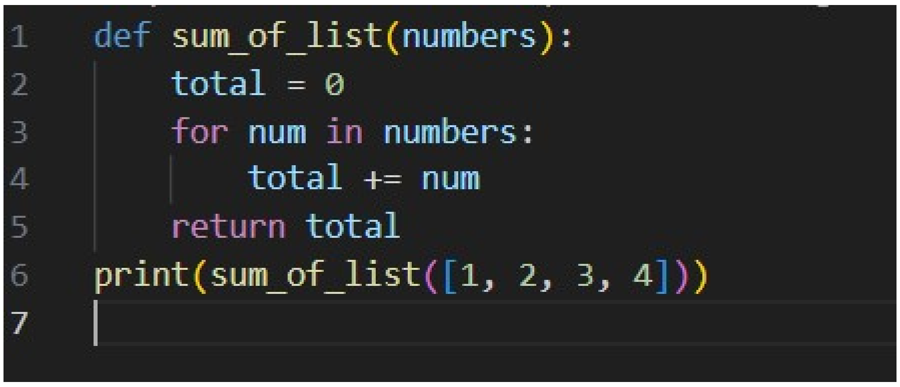


```
Enter an integer: 2
Prime number
```

Task Description-2

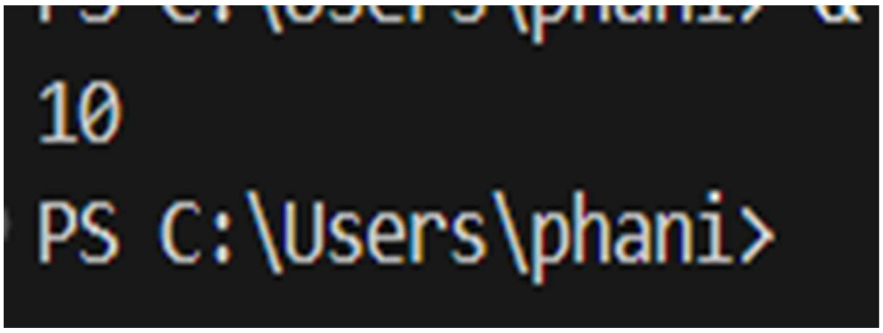
• One-shot: Provide one example: Input: [1, 2, 3, 4], Output: 10 to help AI generate a function that calculates the sum of elements in a list.

Code Screenshot :



```python
def sum_of_list(numbers):
    total = 0
    for num in numbers:
        total += num
    return total
print(sum_of_list([1, 2, 3, 4]))
```

Output:

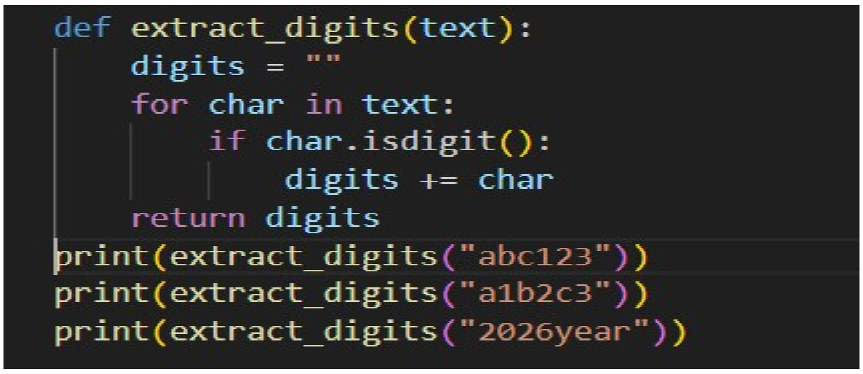


```
10
PS C:\Users\phani>
```

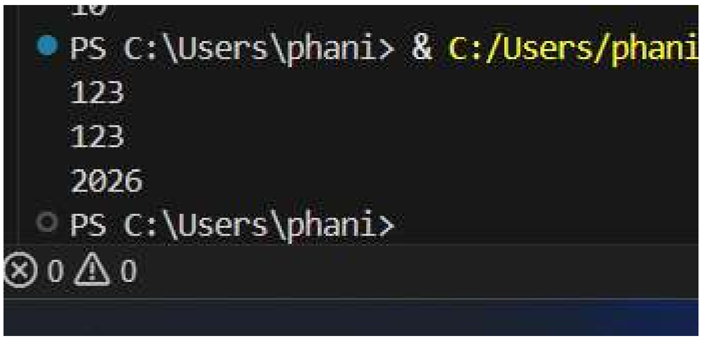
Task Description-3

• Few-shot: Give 2–3 examples to create a function that extracts digits from an alphanumeric string.

Code:


```python
def extract_digits(text):
    digits = ""
    for char in text:
        if char.isdigit():
            digits += char
    return digits
print(extract_digits("abc123"))
print(extract_digits("a1b2c3"))
print(extract_digits("2026year"))
```

Output:

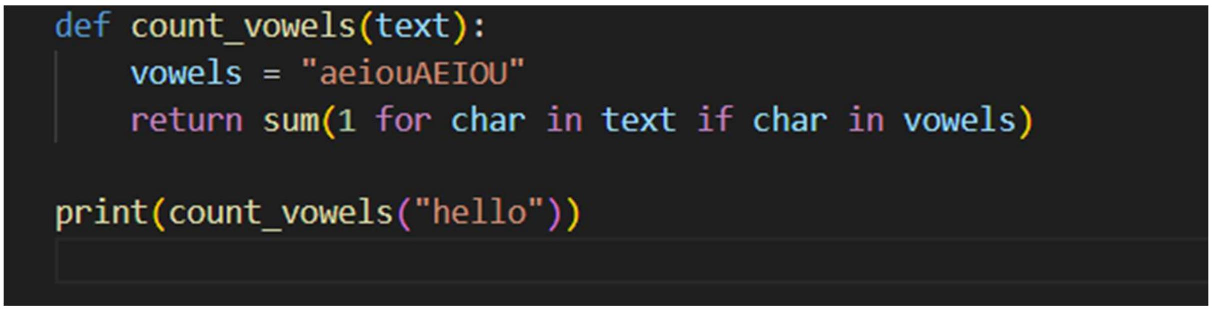

```
PS C:\Users\phani> & C:/Users/phani
123
123
2026
PS C:\Users\phani>
```

Task Description-4:

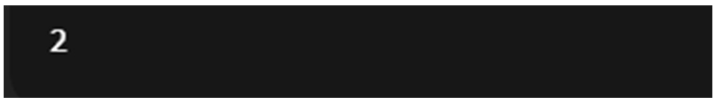
• Compare zero-shot vs few-shot prompting for generating a function that counts the

number of vowels in a string.

Code:


```python
def count_vowels(text):
    vowels = "aeiouAEIOU"
    return sum(1 for char in text if char in vowels)

print(count_vowels("hello"))
```

Output:


```
2
```

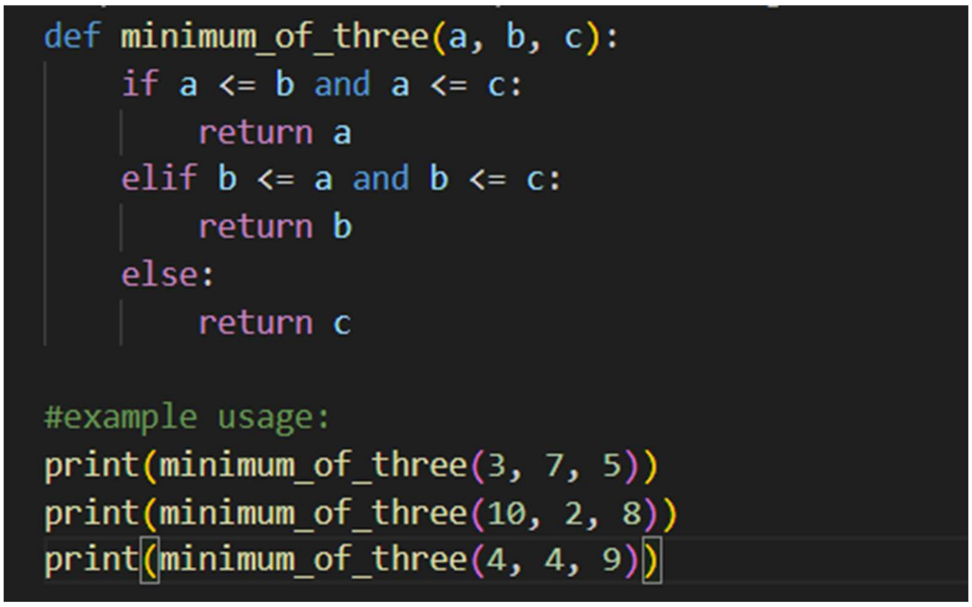
explanation:

The function count_vowels takes a string as input.
It stores all uppercase and lowercase vowels in a variable.
Each character in the string is checked one by one.
If a character is a vowel, it is counted.
Finally, the function returns the total number of vowels.

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.

Code :



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

#example usage:
print(minimum_of_three(3, 7, 5))
print(minimum_of_three(10, 2, 8))
print(minimum_of_three(4, 4, 9))
```

Output:

Input:(3,7,5)→Output:3
Input:(10,2,8)→Output:2
Input: (4, 4, 9) → Output: 4

Explanation: How examples helped the model

The few-shot examples clearly showed how the function should behave in different situations, including normal cases and equal values. By seeing these examples, the model understood that it must compare all three numbers and return the smallest one without using min(). The examples reduced confusion and guided the model to apply proper conditional logic. As a result, the generated function is accurate, consistent, and handles all cases correctly.