LAB ASSIGNMENT-4

Sanjana Bayya
2303A52019
Batch-38

Task Description-1

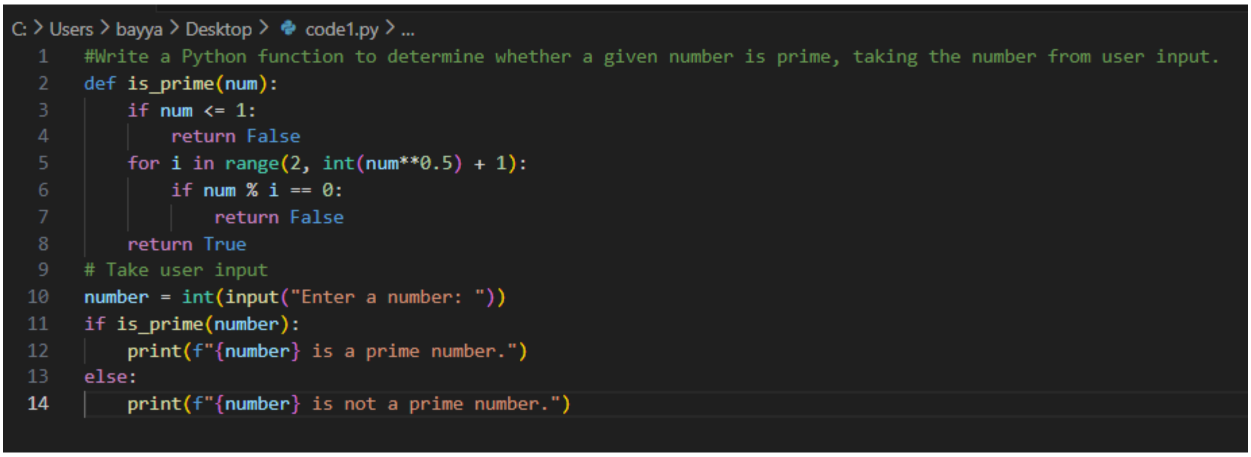
- Zero-shot: Prompt AI with only the instruction. Write a Python function to determine whether a given number is prime

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

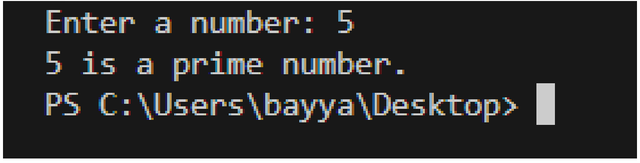
Prompt: Write a Python function to determine whether a given number is prime, taking the number from user input.

Code:



```python
#Write a Python function to determine whether a given number is prime, taking the number from user input.
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True
# Take user input
number = int(input("Enter a number: "))
if is_prime(number):
    print(f"{number} is a prime number.")
else:
    print(f"{number} is not a prime number.")
```

Output:



```
Enter a number: 5
5 is a prime number.
PS C:\Users\bayya\Desktop>
```

Explanation:

This program checks if a number is prime.
The user enters a number n. The function tests divisibility from 2 up to √n.
If no divisor is found, it prints that the number is prime; otherwise, it prints not prime.

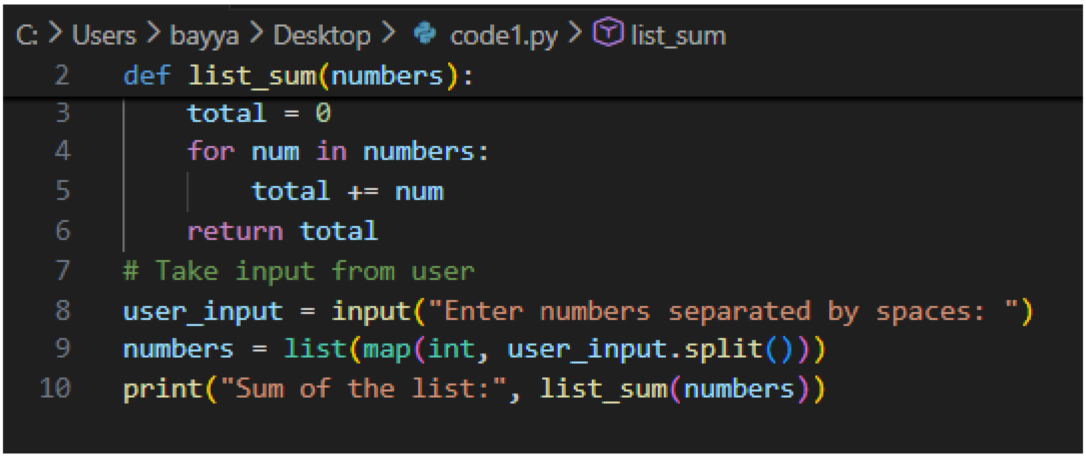
Task Description-2

- 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.

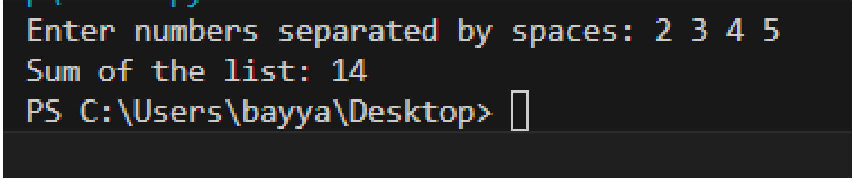
Prompt: Write a Python function that calculates the sum of elements in a list ,for example: Input: [1, 2, 3, 4]

Code:



```
def list_sum(numbers):
    total = 0
    for num in numbers:
        total += num
    return total
# Take input from user
user_input = input("Enter numbers separated by spaces: ")
numbers = list(map(int, user_input.split()))
print("Sum of the list:", list_sum(numbers))
```

Output:



```
Enter numbers separated by spaces: 2 3 4 5
Sum of the list: 14
PS C:\Users\bayya\Desktop> 
```

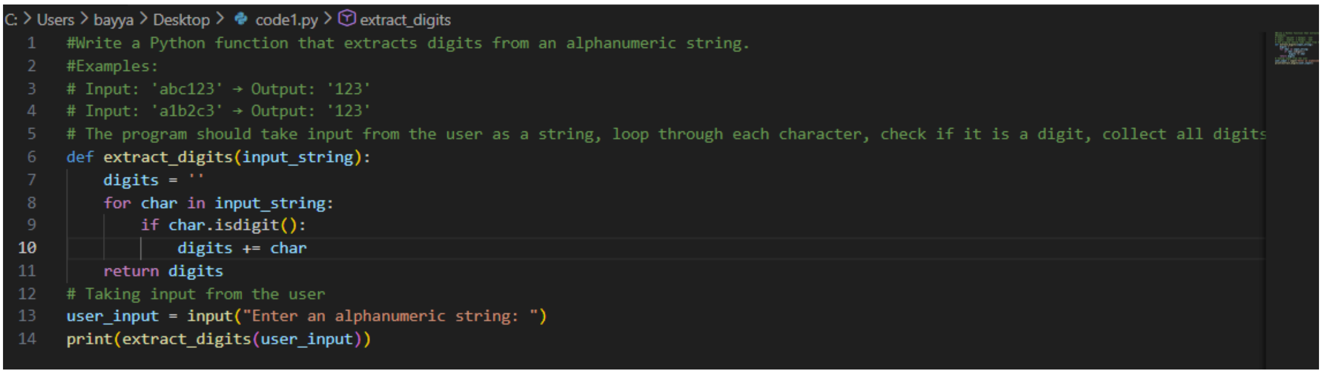
Explanation:
This program calculates the sum of numbers in a list.
The user enters numbers separated by spaces, which are converted into integers.
The function adds them together and prints the total sum.

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.

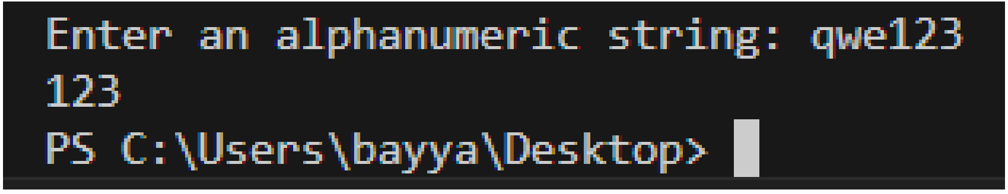
Prompt:
Write a Python function that extracts digits from an alphanumeric string.
Examples:
Input: 'abc123' → Output: '123'
Input: 'a1b2c3' → Output: '123'
The program should take input from the user as a string, loop through each character, check if it is a digit, collect all digits, and print them.

Code:



```python
#Write a Python function that extracts digits from an alphanumeric string.
#Examples:
# Input: 'abc123' → Output: '123'
# Input: 'a1b2c3' → Output: '123'
# The program should take input from the user as a string, loop through each character, check if it is a digit, collect all digits
def extract_digits(input_string):
    digits = ''
    for char in input_string:
        if char.isdigit():
            digits += char
    return digits
# Taking input from the user
user_input = input("Enter an alphanumeric string: ")
print(extract_digits(user_input))
```

Output:



```
Enter an alphanumeric string: qwe123
123
PS C:\Users\bayya\Desktop>
```

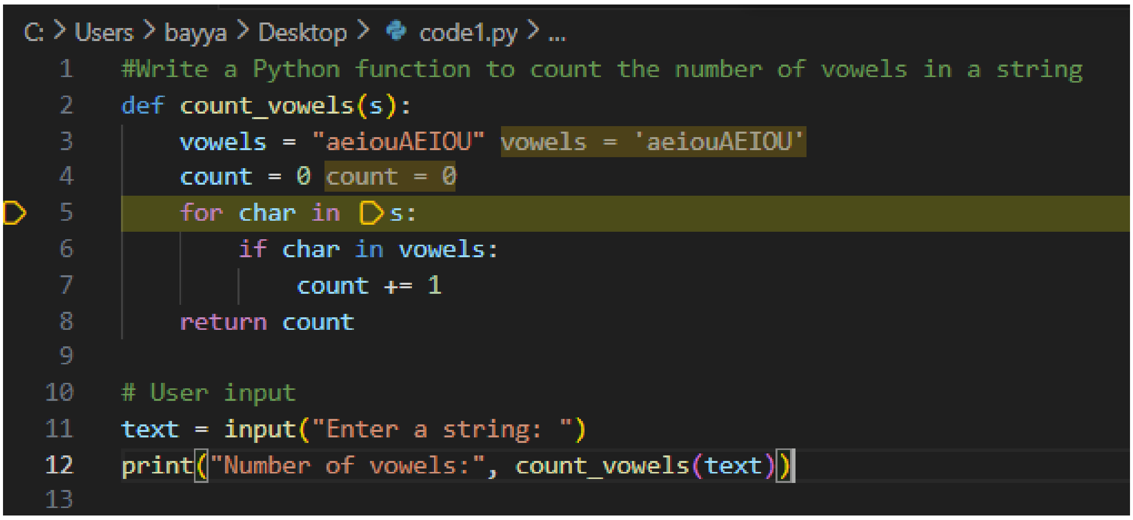
Explanation:
This program extracts digits from an alphanumeric string.
The function checks each character, collects only digits, and returns them as a string.
The user enters a string, and the program prints the digits found.

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 Python function to count the number of vowels in a string.

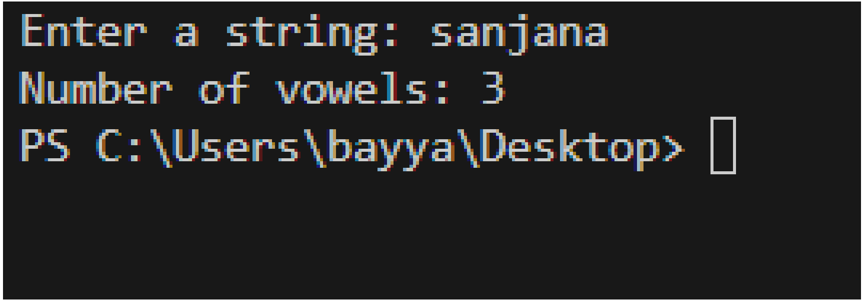
Code:



```
#Write a Python function to count the number of vowels in a string
def count_vowels(s):
    vowels = "aeiouAEIOU" vowels = 'aeiouAEIOU'
    count = 0 count = 0
    for char in s:
        if char in vowels:
            count += 1
    return count

# User input
text = input("Enter a string: ")
print("Number of vowels:", count_vowels(text))
```

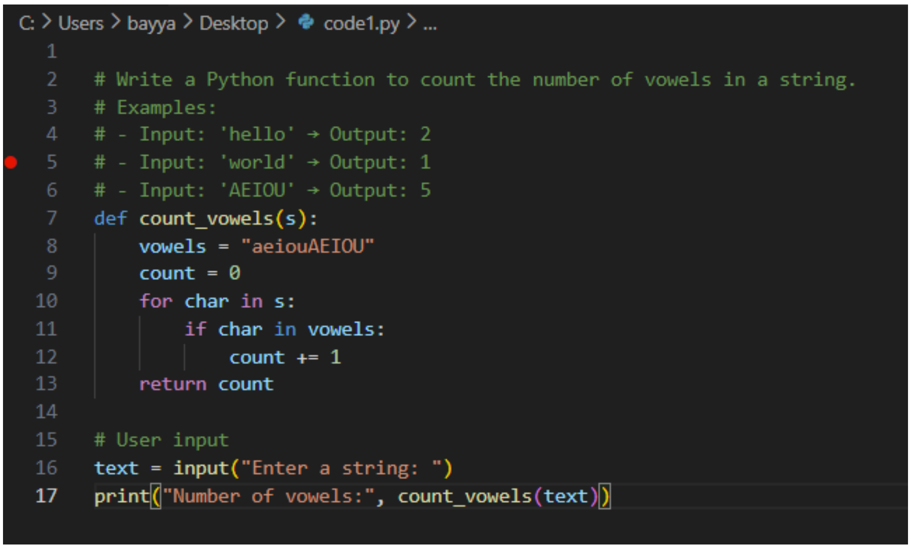
Output: Zero Shot



```
Enter a string: sanjana
Number of vowels: 3
PS C:\Users\bayya\Desktop> 
```

Prompt:
Write a Python function to count the number of vowels in a string.
Examples:
Input: 'hello' → Output: 2
Input: 'world' → Output: 1
Input: 'AEIOU' → Output: 5

Code:

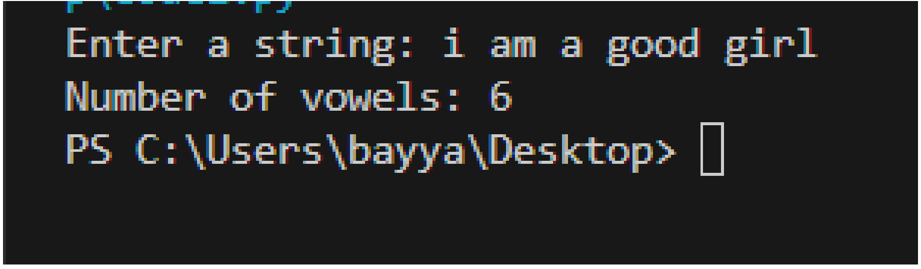


```python

# Write a Python function to count the number of vowels in a string.
# Examples:
# - Input: 'hello' → Output: 2
# - Input: 'world' → Output: 1
# - Input: 'AEIOU' → Output: 5
def count_vowels(s):
    vowels = "aeiouAEIOU"
    count = 0
    for char in s:
        if char in vowels:
            count += 1
    return count

# User input
text = input("Enter a string: ")
print("Number of vowels:", count_vowels(text))
```

Output: Few- Shot



```
Enter a string: i am a good girl
Number of vowels: 6
PS C:\Users\bayya\Desktop> 
```

Explanation:
In zero-shot, the model wrote the function only from the instruction.
In few-shot, the examples guided it to handle different cases more accurately.
Examples make the model more reliable by showing the expected pattern.

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.

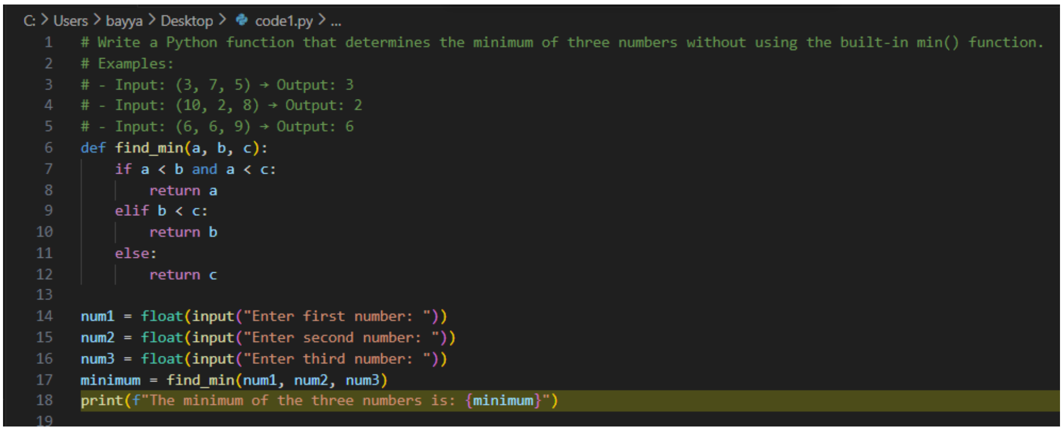
Expected Output-5

- A function that handles all cases with correct logic based on example patterns.

Prompt:

Write a Python function that determines the minimum of three numbers without using the built-in min() function.
Examples:
Input: (3, 7, 5) → Output: 3
Input: (10, 2, 8) → Output: 2
Input: (6, 6, 9) → Output: 6

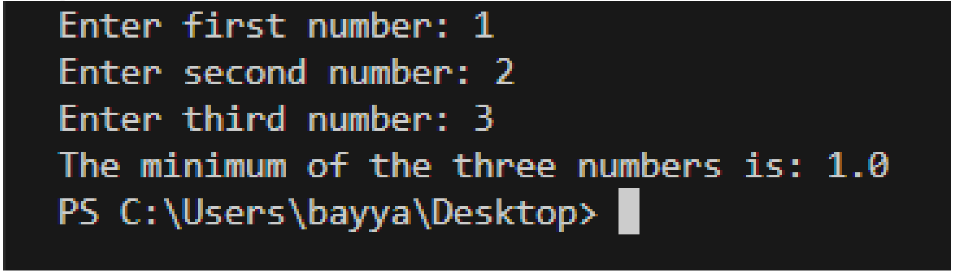
Code:



```python
# Write a Python function that determines the minimum of three numbers without using the built-in min() function.
# Examples:
# - Input: (3, 7, 5) → Output: 3
# - Input: (10, 2, 8) → Output: 2
# - Input: (6, 6, 9) → Output: 6
def find_min(a, b, c):
    if a < b and a < c:
        return a
    elif b < c:
        return b
    else:
        return c

num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))
minimum = find_min(num1, num2, num3)
print(f"The minimum of the three numbers is: {minimum}")
```

Output:



```
Enter first number: 1
Enter second number: 2
Enter third number: 3
The minimum of the three numbers is: 1.0
PS C:\Users\bayya\Desktop>
```

Explanation:
This program finds the smallest of three numbers without using min().
It compares each number step by step to figure out which one is the smallest.
The examples helped the model learn the pattern and apply correct logic for all cases.