LAB ASSIGNMENT 4.2

Name: P.Vaishnavi

HTNO: 2303A51069

Batch – 16

Task 1: Zero-Shot Prompting

Prompt Used: Write a Python function to determine whether a given number is prime.

CODE:

<untitled> *

```
def is_prime(n):
    # Prime numbers are greater than 1
    if n <= 1:
        return False
    # Check divisibility up to sqrt(n)
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False

    return True
num = 29
if is_prime(num):
    print(f"{num} is a prime number")
else:
    print(f"{num} is not a prime number")

```

OUTPUT:

Shell

```
>>> %Run -c $EDITOR_CONTENT
 4 is not a prime number

>>>
```

Code Explanation:

- The function `is_prime(n)` checks whether a number is prime.
- If the number is less than or equal to 1, it returns `False` because prime numbers start from 2.
- A loop runs from 2 up to the square root of the number to check divisibility.
- If the number is divisible by any value in this range, it is not prime.
- If no divisor is found, the function returns `True`, confirming the number is prime.

Task 2: One-Shot Prompting

Prompt Used:

Input: [1, 2, 3, 4]

Output: 10

Write a Python function to calculate the sum of elements in a list**.**

CODE:

<untitled> *

```
# Function to calculate sum of elements in a list
def calculate_sum(lst):
    total = 0

    # Add each element to total
    for num in lst:
        total += num

    return total

# Sample test case
print(calculate_sum([1, 2, 3, 4]))
```

OUTPUT:

Code Explanation:

- The function initializes a variable to store the sum.
- A loop iterates through the list.
- Each element is added to the total.
- The final sum is returned.

Task 3: Few-Shot Prompting

Prompt Used:

Example 1:

Input: "a1b2c3"

Output: "123"

Example 2:

Input: "abc456"

Output: "456"

Example 3:

Input: "9x8y7"

Output: "987"

Write a Python function to extract digits from an alphanumeric string.

CODE:

<untitled> *

```python
# Function to extract digits from an alphanumeric string
def extract_digits(text):
    digits = ""

    # Loop through each character
    for ch in text:
        if ch.isdigit():
            digits += ch

    return digits

# Sample test cases
print(extract_digits("a1b2c3"))
print(extract_digits("abc456"))
```

OUTPUT:

Shell

```
>>> %Run -c $EDITOR_CONTENT
123
456

>>>
```

Code Explanation:

- An empty string stores digits.
- Each character is checked using isdigit().
- Digits are appended to the result.
- The function returns only numeric characters.

Task 4: Zero-Shot vs Few-Shot Prompting

Zero-Shot Prompt:

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

Zero-Shot Code:

<untitled> *

```python
# Function to count vowels in a string
def count_vowels(s):
    count = 0

    for ch in s:
        if ch in "aeiou":
            count += 1

    return count

print(count_vowels("hello"))

```

OUTPUT:

```
>>> %Run -c $EDITOR_CONTENT
 2

>>>
```

Few-Shot Prompt:

Input: "education"

Output: 5

Write a Python function to count vowels in a string.

Few-Shot Code:

<untitled> *

```
# Improved function to count vowels (both cases)
def count_vowels(s):
    vowels = "aeiouAEIOU"
    return sum(1 for ch in s if ch in vowels)

print(count_vowels("Education"))

```

OUTPUT:

```
>>> %Run -c $EDITOR_CON
 5

>>>
```

Code Explanation:

- Zero-shot handles only lowercase vowels.
- Few-shot handles both uppercase and lowercase.
- Examples improved correctness and optimisation.

Task 5: Few-Shot Prompting

Prompt Used:

Example 1:

Input: 3, 5, 7

Output: 3

Example 2:

Input: 10, 2, 8

Output: 2

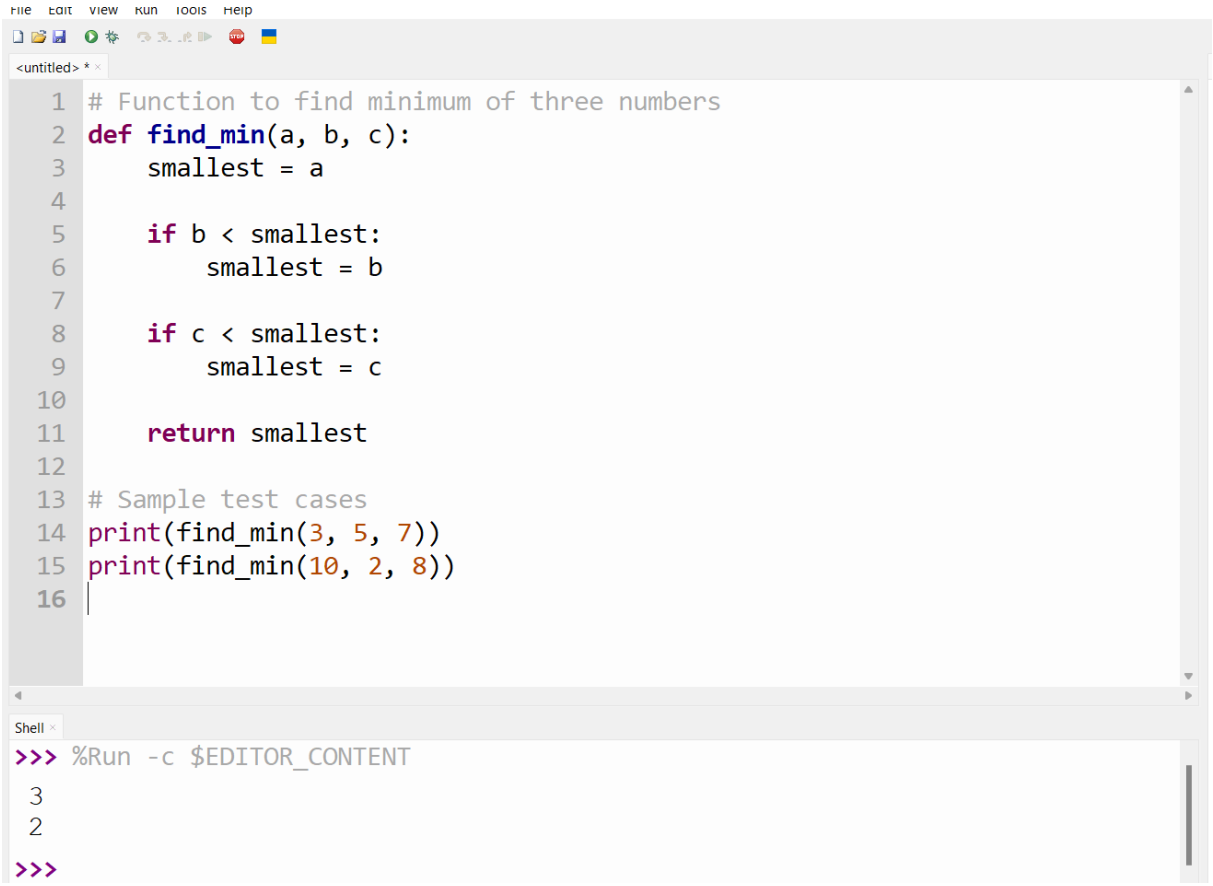
Example 3:

Input: 4, 4, 6

Output: 4

Write a Python function to find the minimum of three numbers without using min().

CODE:



Code Explanation:

- First value is assumed as the minimum.
- It is compared with the second and third values.
- The smallest value is updated accordingly.
- Final minimum value is returned.