AI ASSISTANT CODING

ASSIGNMENT-4

Name: Kommu Madhupriya

Hallticket:2303A51583

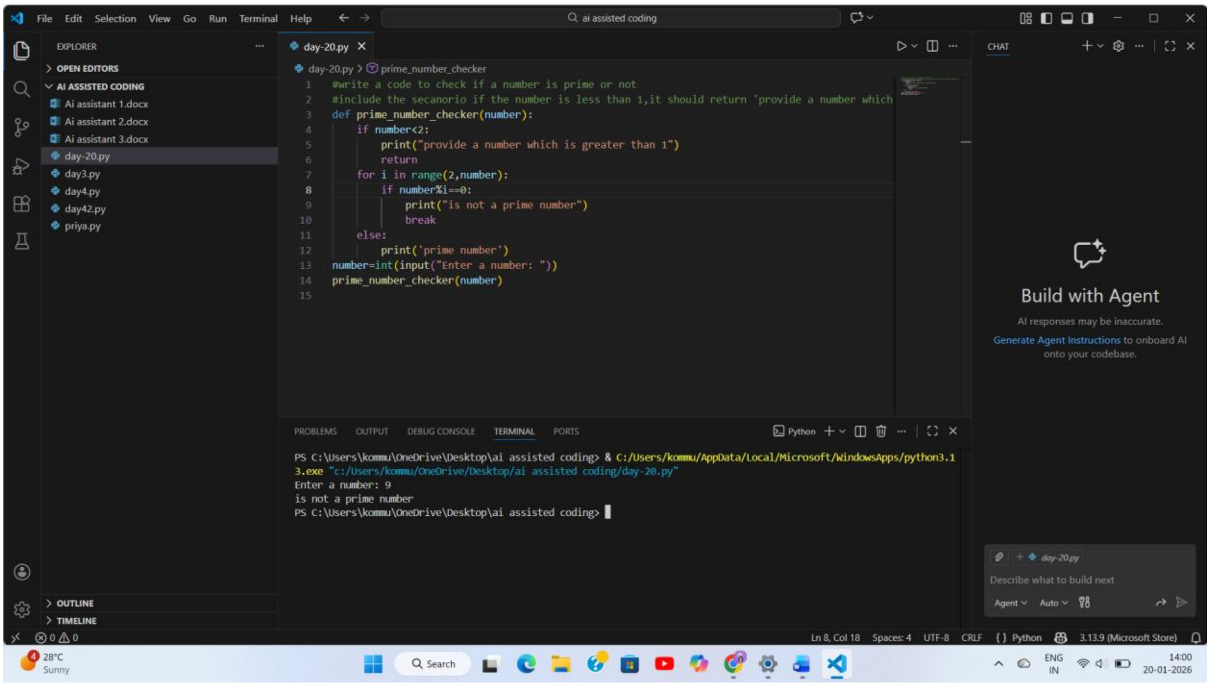
Batch:22

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.


File Edit Selection View Go Run Terminal Help
ai assisted coding
EXPLORER
OPEN EDITORS
AI ASSISTED CODING
Ai assistant 1.docx
Ai assistant 2.docx
Ai assistant 3.docx
day-20.py
day3.py
day4.py
day42.py
priya.py
OUTLINE
TIMELINE
day-20.py > prime_number_checker
CHAT
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
Describe what to build next
Agent Auto
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
Ln 8, Col 18 Spaces: 4 UTF-8 CRLF {} Python 3.13.9 (Microsoft Store)
28°C Sunny
Search
ENG IN
14:00 20-01-2026

```python
#write a code to check if a number is prime or not
#include the secanorio if the number is less than 1,it should return 'provide a number which
def prime_number_checker(number):
    if number<2:
        print("provide a number which is greater than 1")
        return
    for i in range(2,number):
        if number%i==0:
            print("is not a prime number")
            break
    else:
        print('prime number')
number=int(input("Enter a number: "))
prime_number_checker(number)
```

```
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.1
3.exe "c:/Users/kommu/OneDrive/Desktop/ai assisted coding/day-20.py"
Enter a number: 9
is not a prime number
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> 
```

Explanation:

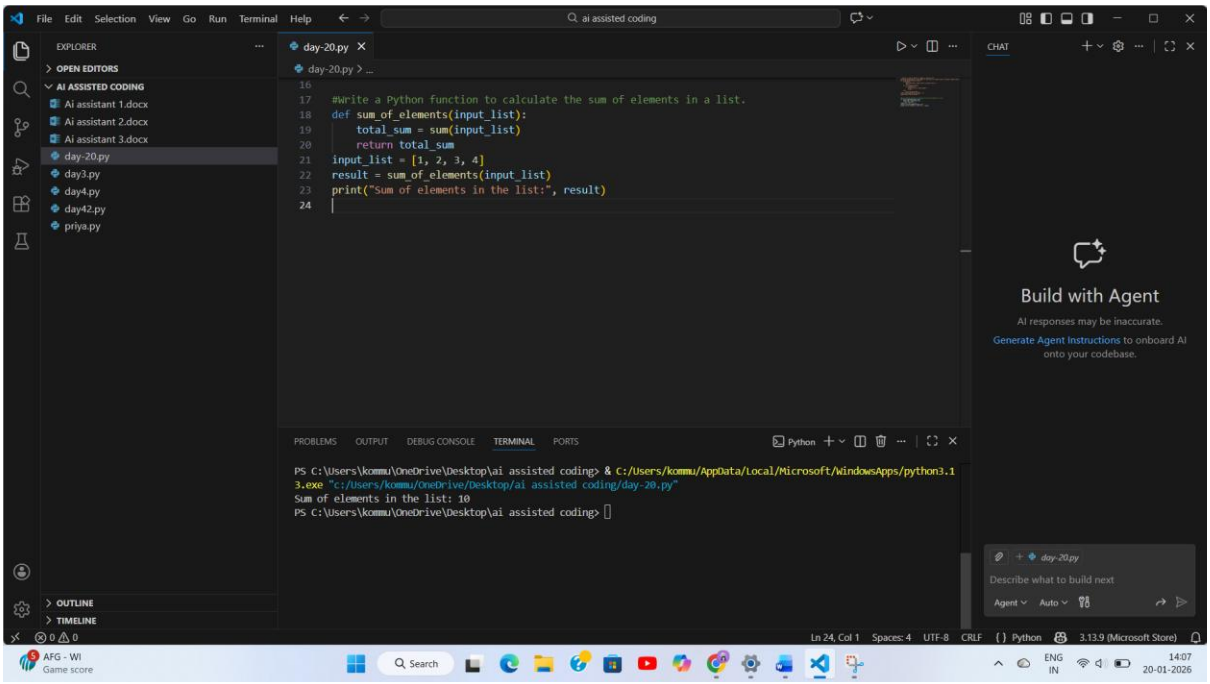
In this program, we check whether a given number is a prime number or not. First, we check if the number is less than or equal to 1. If it is, then it is not a prime number. After that, we use a loop to divide the number by all values starting from 2 up to one less than the number. If the number is divisible by any of these values, it means the number has more than two factors, so it is not prime. If the loop finishes without finding any divisor, then the number is a 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.

Expected Output-2

- A correct conversion function guided by the single example.



```python
#Write a Python function to calculate the sum of elements in a list.
def sum_of_elements(input_list):
    total_sum = sum(input_list)
    return total_sum
input_list = [1, 2, 3, 4]
result = sum_of_elements(input_list)
print("Sum of elements in the list:", result)
```

```
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.1
3.exe "c:/Users/kommu/OneDrive/Desktop/ai assisted coding/day-20.py"
Sum of elements in the list: 10
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> 
```

Build with Agent

AI responses may be inaccurate.

Generate Agent Instructions to onboard AI onto your codebase.

Explanation:

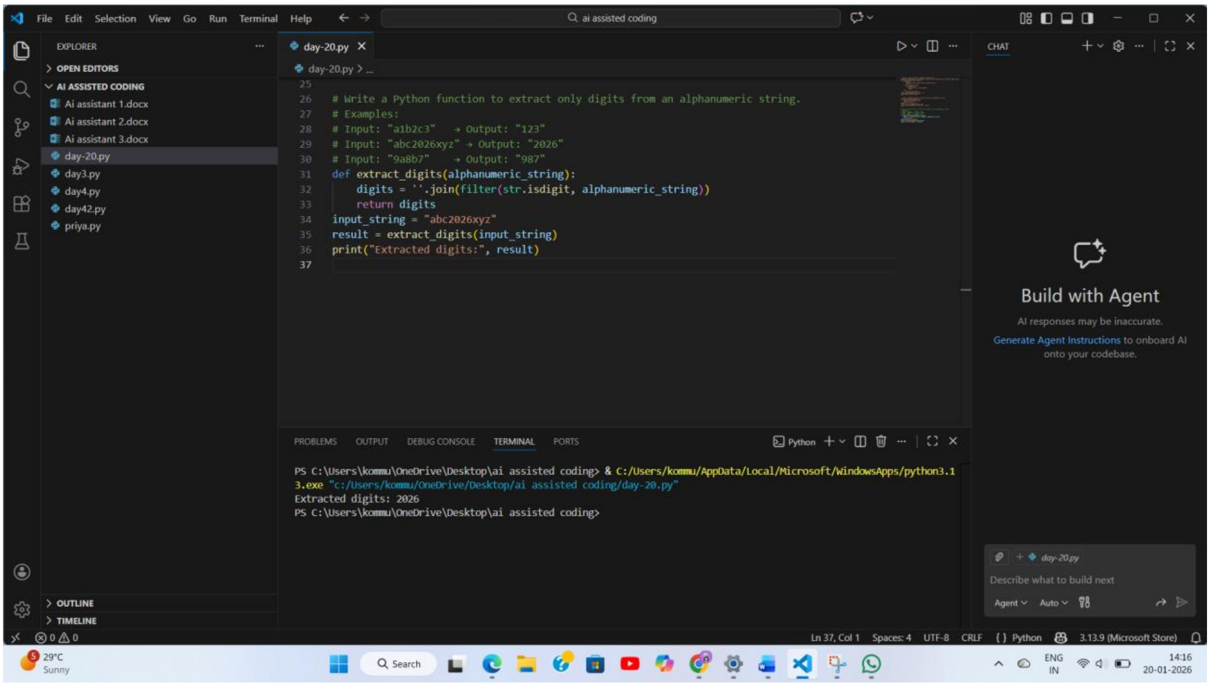
In this program, we calculate the sum of all elements in a list. The given example shows that all the numbers in the list should be added together to get the final result. First, we initialize a variable total with 0. Then, we use a loop to go through each number in the list and add it to total. After the loop ends, the function returns the total sum of the list elements.

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.



```python
# Write a Python function to extract only digits from an alphanumeric string.
# Examples:
# Input: "a1b2c3"    → Output: "123"
# Input: "abc2026xyz" → Output: "2026"
# Input: "9a8b7"     → Output: "987"
def extract_digits(alphanumeric_string):
    digits = ''.join(filter(str.isdigit, alphanumeric_string))
    return digits
input_string = "abc2026xyz"
result = extract_digits(input_string)
print("Extracted digits:", result)
```

```
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.1
3.exe "c:/Users/kommu/OneDrive/Desktop/ai assisted coding/day-20.py"
Extracted digits: 2026
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>
```

Explanation:

This program is used to extract only the digits from an alphanumeric string. First, an empty string named digits is created to store the numbers found in the input string. Then, a loop checks each character in the string one by one. If the character is a digit, it is added to the digits string using isdigit(). Characters that are not digits are ignored. After checking all the characters, the function returns a string that contains only the digits in the same order as they appear in the original string.

Task Description-4

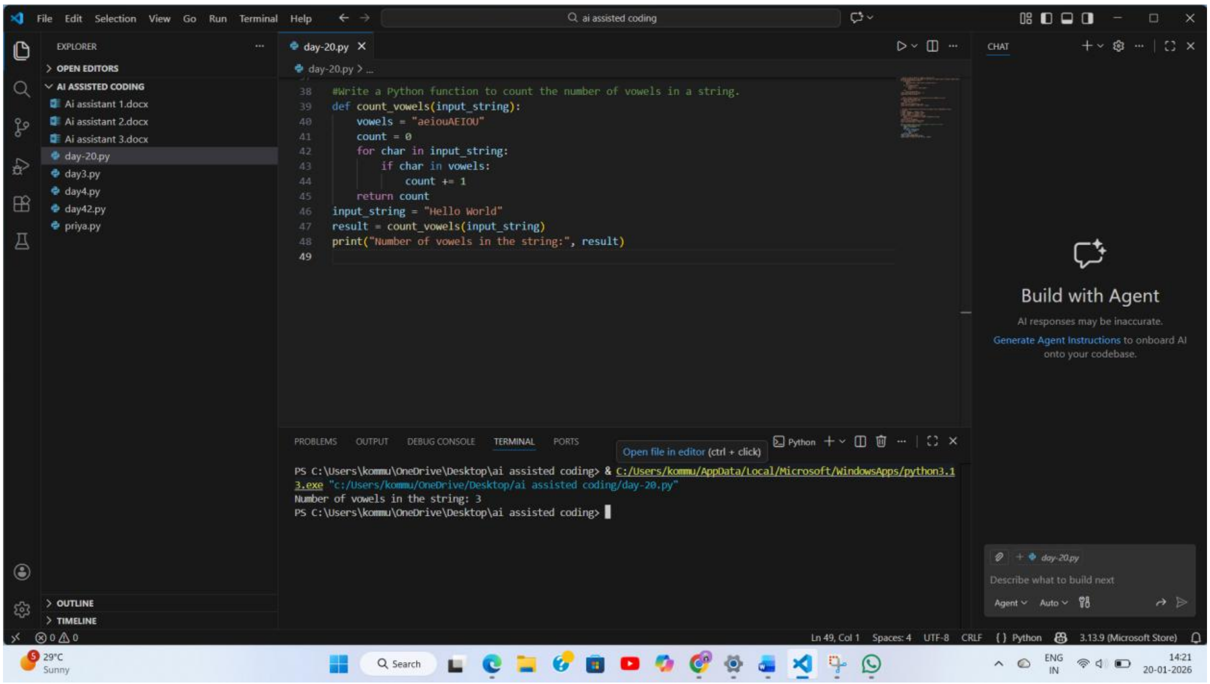
- Compare zero-shot vs few-shot prompting for generating a function that counts the

number of vowels in a string.

Expected Output-4

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

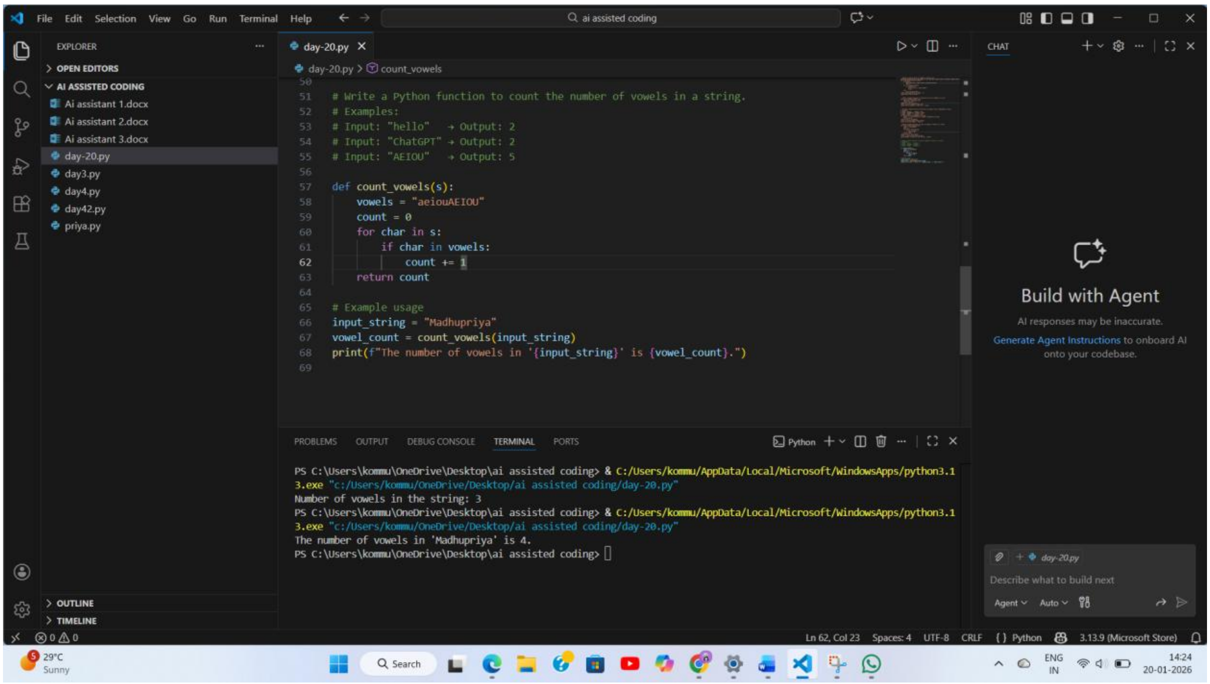
ZERO SHOT



```python
#Write a Python function to count the number of vowels in a string.
def count_vowels(input_string):
    vowels = "aeiouAEIOU"
    count = 0
    for char in input_string:
        if char in vowels:
            count += 1
    return count
input_string = "Hello World"
result = count_vowels(input_string)
print("Number of vowels in the string:", result)
```

```
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.1
3.exe "c:/Users/kommu/OneDrive/Desktop/ai assisted coding/day-20.py"
Number of vowels in the string: 3
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>
```

FEW SHOT:



```python
# Write a Python function to count the number of vowels in a string.
# Examples:
# Input: "hello"   → Output: 2
# Input: "ChatGPT" → Output: 2
# Input: "AEIOU"   → Output: 5

def count_vowels(s):
    vowels = "aeiouAEIOU"
    count = 0
    for char in s:
        if char in vowels:
            count += 1
    return count

# Example usage
input_string = "Madhupriya"
vowel_count = count_vowels(input_string)
print(f"The number of vowels in '{input_string}' is {vowel_count}.")
```

```
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.1
3.exe "c:/Users/kommu/OneDrive/Desktop/ai assisted coding/day-20.py"
Number of vowels in the string: 3
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.1
3.exe "c:/Users/kommu/OneDrive/Desktop/ai assisted coding/day-20.py"
The number of vowels in 'Madhupriya' is 4.
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>
```

Explanation:

In zero-shot prompting, the AI receives only the instruction, so it generates a basic solution using a simple loop to count vowels. The logic is correct but straightforward. In few-shot prompting, the given examples help the AI understand different cases, such as strings with many vowels and strings with no vowels at all. Because of these examples, the AI produces a more concise and efficient solution. This shows that providing examples improves the quality, clarity, and confidence of the generated code.

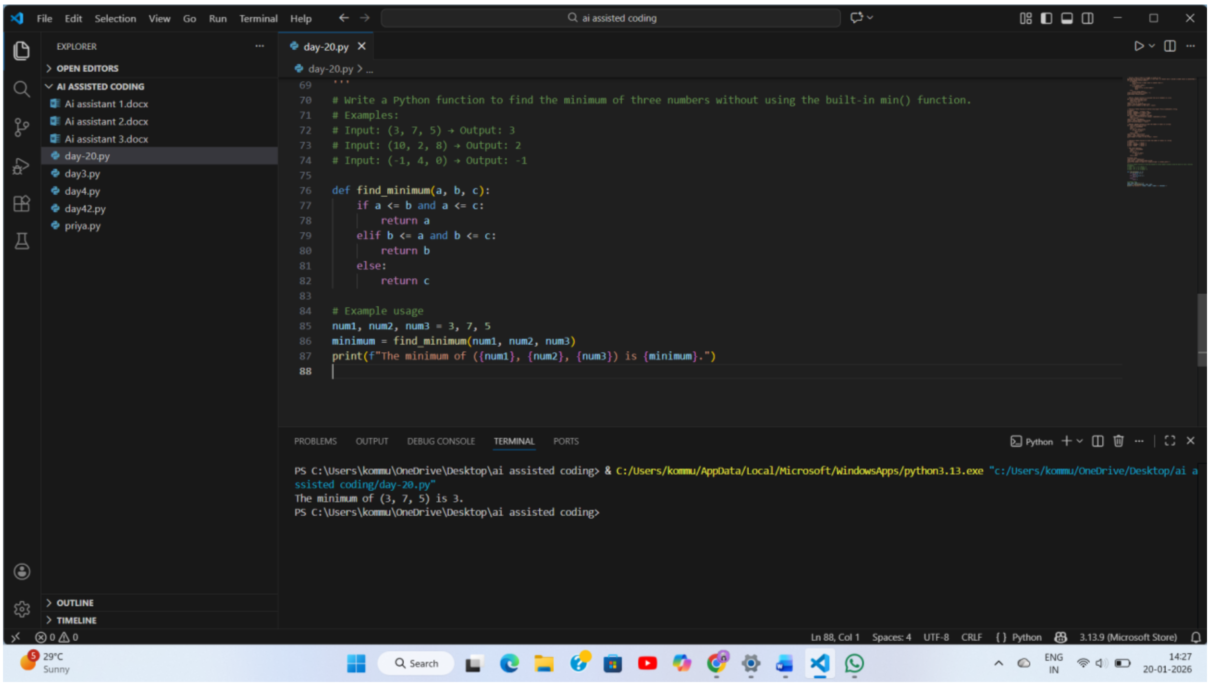
Task Description-5

- 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.



```python
'''
# Write a Python function to find 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: (-1, 4, 0) → Output: -1

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
num1, num2, num3 = 3, 7, 5
minimum = find_minimum(num1, num2, num3)
print(f"The minimum of ({num1}, {num2}, {num3}) is {minimum}.")
```

```
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding> & C:/Users/kommu/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/kommu/OneDrive/Desktop/ai a
ssisted coding/day-20.py"
The minimum of (3, 7, 5) is 3.
PS C:\Users\kommu\OneDrive\Desktop\ai assisted coding>
```

Explanation:

In this program, we find the smallest number among three given numbers without using the min() function. The examples show that the function should correctly handle different values, including equal numbers. First, the function compares the first number with the other two. If it is smaller than or equal to both, it is returned as the minimum. If not, the second number is compared with the other two. If it is the smallest, it is returned. Otherwise, the third number is returned. This logic ensures that all possible cases are handled correctly.