

# ASSIGNMENT

NAME: B. Akhira Nandhini

HTNO:2303A51516

BATCH:22

---

## AI ASSISTANT CODING

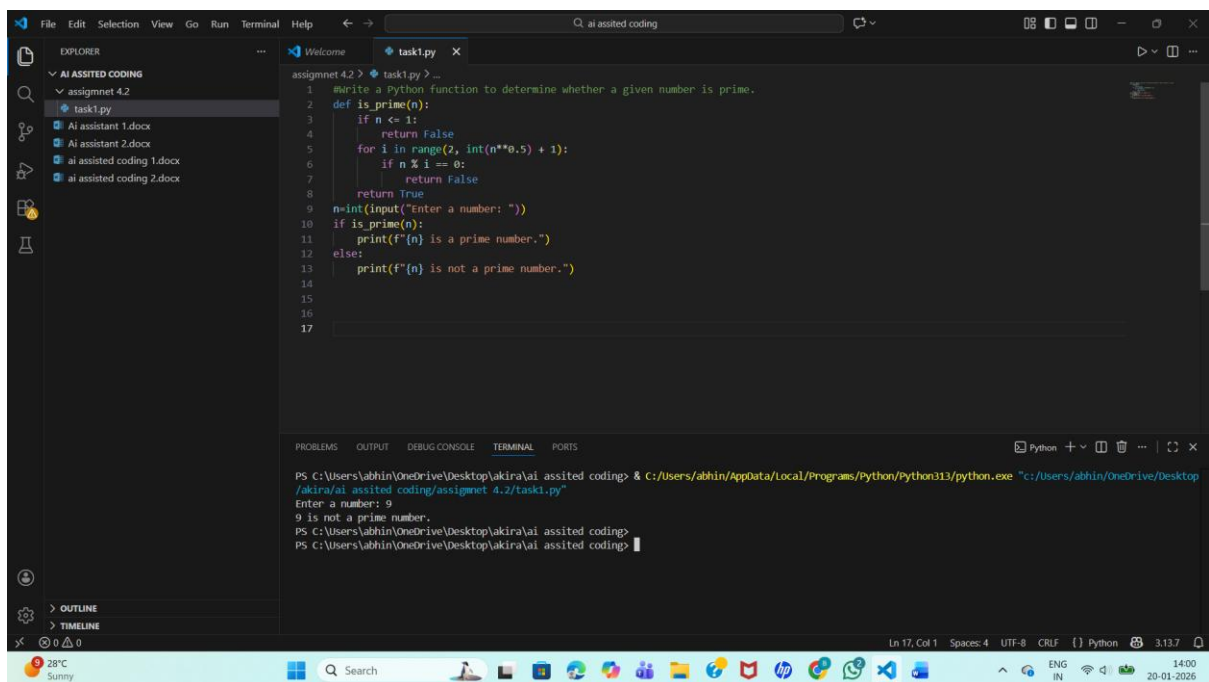
### Task Description-1

- 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

### INPUT and OUTPUT:



The screenshot shows a Visual Studio Code editor with a Python file named `task1.py`. The code defines a function `is_prime(n)` that checks if a number is prime. It handles the case where `n` is less than or equal to 1, and then checks for divisibility by numbers from 2 to  $\sqrt{n}$ . The script prompts the user to enter a number and prints the result.

```
1 #Write a Python function to determine whether a given number is prime.
2 def is_prime(n):
3     if n <= 1:
4         return False
5     for i in range(2, int(n**0.5) + 1):
6         if n % i == 0:
7             return False
8     return True
9 n=int(input("Enter a number: "))
10 if is_prime(n):
11     print(f"{n} is a prime number.")
12 else:
13     print(f"{n} is not a prime number.")
14
15
16
17
```

The terminal output shows the script being executed with the input 9, resulting in the message "9 is not a prime number.".

```
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding> & C:\Users\abhin\AppData\Local\Programs\Python\Python313\python.exe "C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding\assignment 4.2\task1.py"
Enter a number: 9
9 is not a prime number.
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding>
```

### Explanation:

- A number that is less than or equal to 1 cannot be a prime number, so first we check for that condition.

- Then, to find whether the number is prime or not, we divide it by numbers starting from 2 up to the square root of the given number.
- If the number gets divided by any of these values, it means it has factors and is not a prime number.
- If no divisor is found in this range, then the number has only two factors (1 and itself), so it is a prime number.

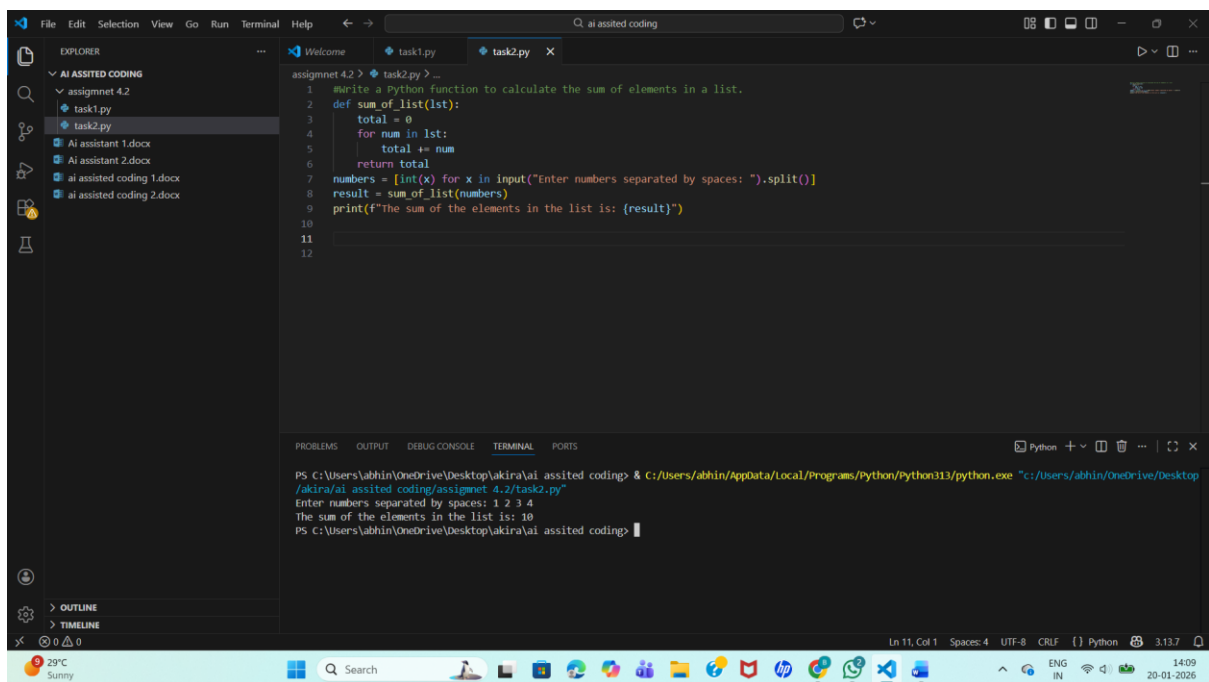
## Task Description-2

- 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.

## INPUT and OUTPUT:



```

1 #Write a Python function to calculate the sum of elements in a list.
2 def sum_of_list(list):
3     total = 0
4     for num in list:
5         total += num
6     return total
7 numbers = [int(x) for x in input("Enter numbers separated by spaces: ").split()]
8 result = sum_of_list(numbers)
9 print(f"The sum of the elements in the list is: {result}")
10
11
12

```

```

PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding> & C:/Users/abhin/AppData/Local/Programs/Python/Python313/python.exe "C:/Users/abhin/OneDrive/Desktop/akira/ai assisted coding/assignmet 4.2/task2.py"
Enter numbers separated by spaces: 1 2 3 4
The sum of the elements in the list is: 10
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding>

```

## EXPLANATION:

- In this task, the function is used to find the sum of all elements present in a list.
- First, a variable is initialized to store the total sum.
- Then, the function goes through each element in the list one by one using a loop and adds each value to the total.

- After all the elements are added, the final sum is returned as the output.
- This approach is simple and clearly shows how the sum of a list can be calculated using basic logic without directly using any built-in function.

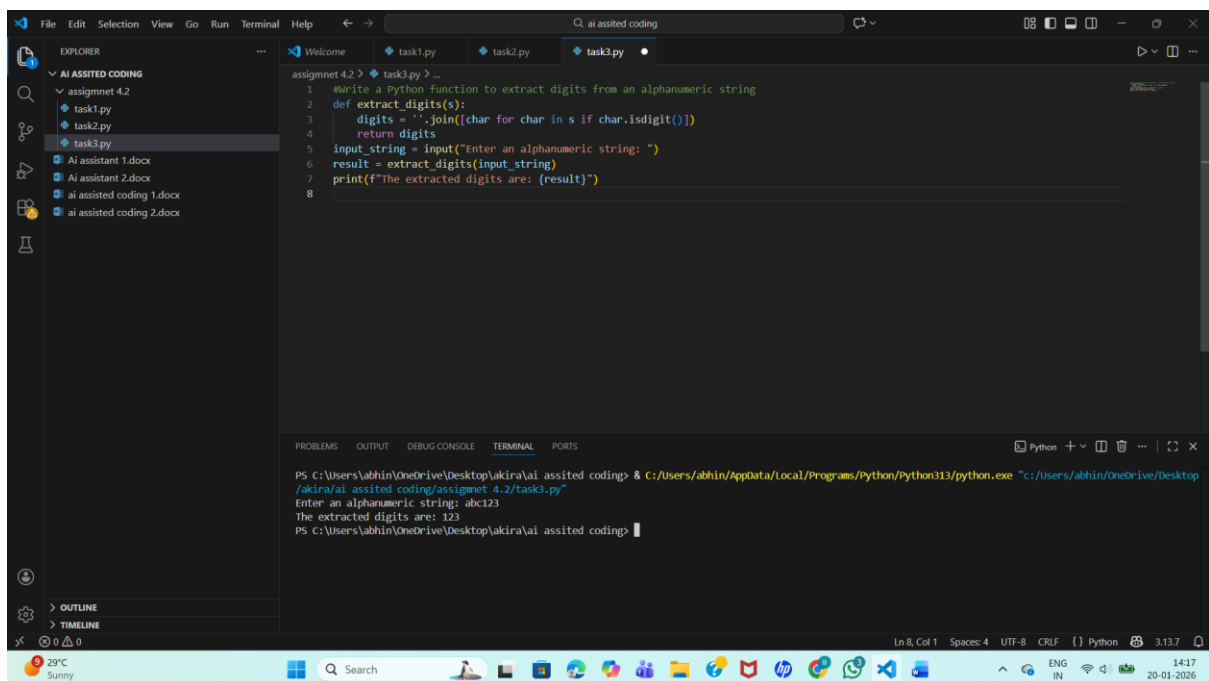
### Task Description-3

- 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.

### INPUT AND OUTPUT:



The screenshot shows a Visual Studio Code editor with a Python file named `task3.py` open. The code defines a function `extract_digits(s)` that iterates through each character in the string `s`. If the character is a digit (checked using `char.isdigit()`), it is added to a new string `digits`. After the loop, the function returns the `digits` string. The script prompts the user to enter an alphanumeric string and prints the extracted digits.

```

1 #write a Python function to extract digits from an alphanumeric string
2 def extract_digits(s):
3     digits = ""
4     for char in s:
5         if char.isdigit():
6             digits += char
7     return digits
8
9 input_string = input("Enter an alphanumeric string: ")
10 result = extract_digits(input_string)
11 print(f"The extracted digits are: {result}")

```

The terminal output shows the execution of the script. The user enters `abc123`, and the program outputs `The extracted digits are: 123`.

```

PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding> & C:/Users/abhin/AppData/Local/Programs/Python/Python313/python.exe "C:/Users/abhin/OneDrive/Desktop/akira/ai assisted coding/assignment 4.2/task3.py"
Enter an alphanumeric string: abc123
The extracted digits are: 123
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding>

```

### EXPLANATION:

- In this task, the function is designed to extract only the digits from an alphanumeric string.
- The function checks each character of the given string one by one.
- If the character is a number, it is added to a new string, and if it is a letter or any other symbol, it is ignored.
- By the end of the loop, the new string contains only the digits from the original input.

- This method makes it easy to separate numbers from text and works correctly for any alphanumeric string.

## Task Description-4

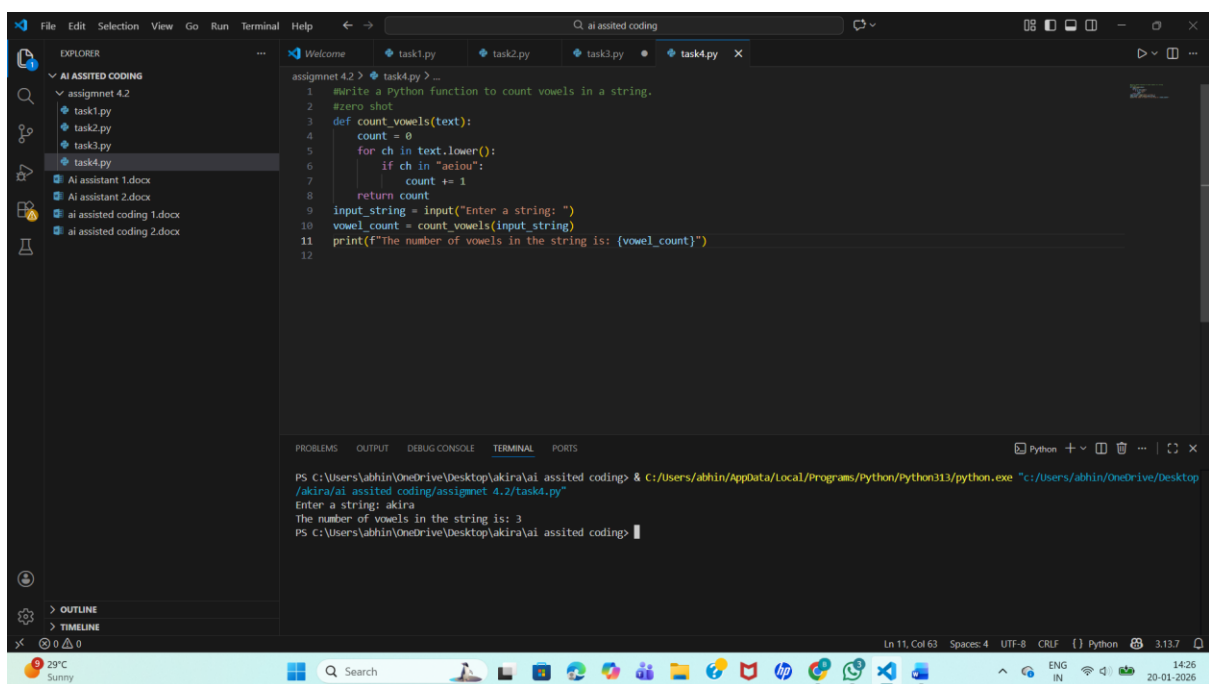
- Compare zero-shot vs few-shot prompting for generating a function that counts the number of vowels in a string.

## Expected Output-4

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

## INPUT AND OUTPUT:

### Zero shot:



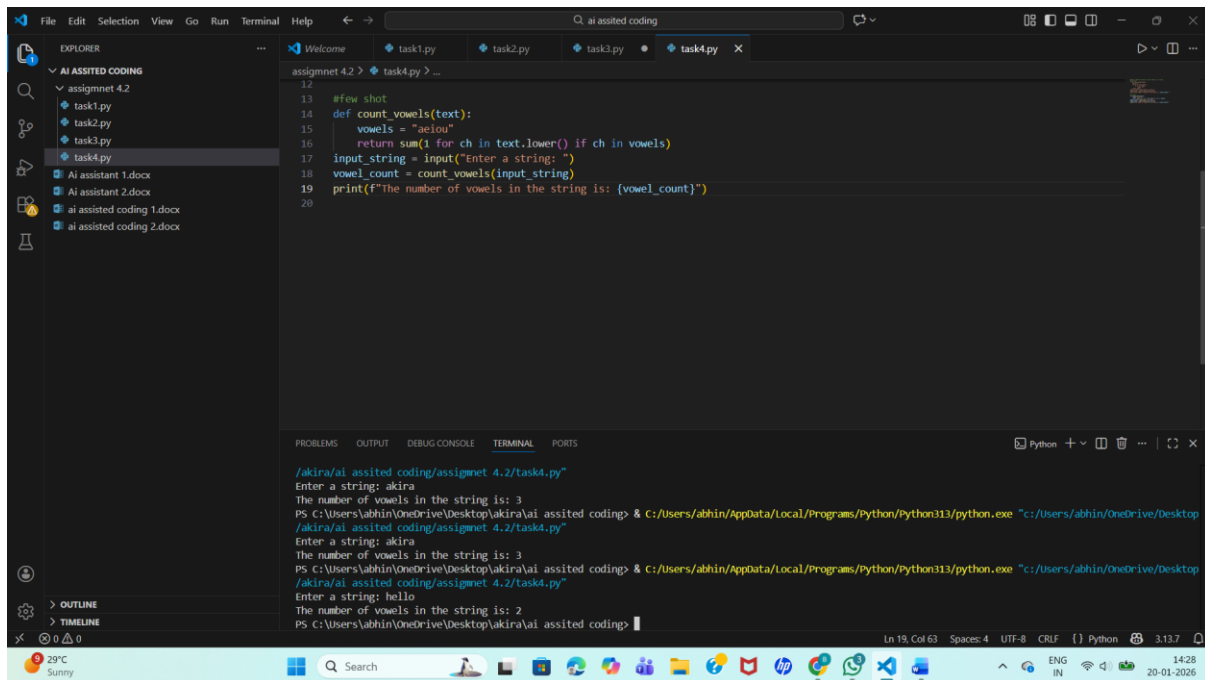
The screenshot shows a Visual Studio Code editor window with a Python file named `task4.py`. The code defines a function `count_vowels` that takes a string `text` and returns the number of vowels. It uses a loop to iterate over each character in the string, converting it to lowercase and checking if it is a vowel (a, e, i, o, u). The function is then called with the input string "akira", and the result is printed.

```
1 #Write a Python function to count vowels in a string.
2 #zero shot
3 def count_vowels(text):
4     count = 0
5     for ch in text.lower():
6         if ch in "aeiou":
7             count += 1
8     return count
9 input_string = input("Enter a string: ")
10 vowel_count = count_vowels(input_string)
11 print(f"The number of vowels in the string is: {vowel_count}")
12
```

The terminal output shows the execution of the script. The user enters "akira" when prompted, and the program outputs "The number of vowels in the string is: 3".

```
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding> & C:/Users/abhin/AppData/Local/Programs/Python/Python313/python.exe "C:/Users/abhin/OneDrive/Desktop/akira/ai assisted coding/assignmet 4.2/task4.py"
Enter a string: akira
The number of vowels in the string is: 3
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding>
```

### Few shot:



```
12
13 #few shot
14 def count_vowels(text):
15     vowels = "aeiou"
16     return sum(1 for ch in text.lower() if ch in vowels)
17 input_string = input("Enter a string: ")
18 vowel_count = count_vowels(input_string)
19 print(f"The number of vowels in the string is: {vowel_count}")
20
```

```
/akira/ai assisted coding/assignmet 4.2/task4.py
Enter a string: akira
The number of vowels in the string is: 3
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding> & C:/Users/abhin/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/abhin/OneDrive/Desktop/akira/ai assisted coding/assignmet 4.2/task4.py"
Enter a string: akira
The number of vowels in the string is: 3
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding> & C:/Users/abhin/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/abhin/OneDrive/Desktop/akira/ai assisted coding/assignmet 4.2/task4.py"
Enter a string: hello
The number of vowels in the string is: 2
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding>
```

## EXPLANATION:

- In this task, zero-shot and few-shot prompting are compared using a function that counts the number of vowels in a string.
- In the zero-shot approach, only the instruction is given, so the function is written using basic logic with loops and conditions.
- In the few-shot approach, examples are provided, which help in understanding the pattern more clearly and allow the function to be written in a more optimized and concise way.
- By comparing both outputs, it is observed that while both methods give correct results, few-shot prompting improves clarity, efficiency, and overall quality of the generated code because the examples guide the model better.

## Task Description-5

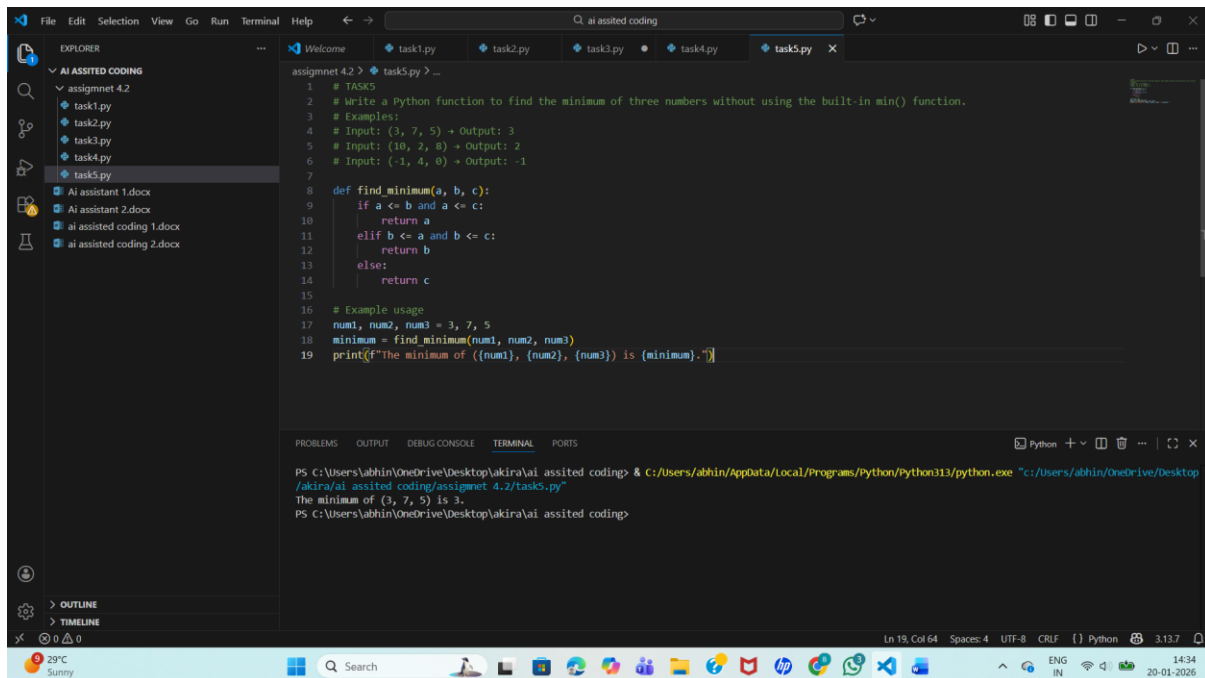
- Use few-shot prompting with 3 sample inputs to generate a function that determines

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.

## INPUT AND OUTPUT:



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'AI ASSISTED CODING' with several files, including 'task5.py'. The code editor displays a Python script named 'task5.py' with the following content:

```
1 # TASKS
2 # Write a Python function to find the minimum of three numbers without using the built-in min() function.
3 # Examples:
4 # Input: (3, 7, 5) -> Output: 3
5 # Input: (10, 2, 8) -> Output: 2
6 # Input: (-1, 4, 0) -> Output: -1
7
8 def find_minimum(a, b, c):
9     if a <= b and a <= c:
10         return a
11     elif b <= a and b <= c:
12         return b
13     else:
14         return c
15
16 # Example usage
17 num1, num2, num3 = 3, 7, 5
18 minimum = find_minimum(num1, num2, num3)
19 print(f"The minimum of ({num1}, {num2}, {num3}) is {minimum}.")
```

Below the code editor, the terminal window shows the execution of the script:

```
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding> & C:/Users/abhin/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/abhin/OneDrive/Desktop/akira/ai assisted coding/assignment 4.2/task5.py"
The minimum of (3, 7, 5) is 3.
PS C:\Users\abhin\OneDrive\Desktop\akira\ai assisted coding>
```

## EXPLANATION:

- In this task, the function is written to find the minimum of three numbers without using the built-in `min()` function.
- The function compares the three values using conditional statements. First, it checks whether the first number is smaller than or equal to the other two.
- If not, it then checks the second number in the same way. If neither of these conditions is true, the third number is considered the smallest.
- This logic ensures that all possible cases are handled correctly, and the correct minimum value is returned based on the given examples.