

Harika Jupaka

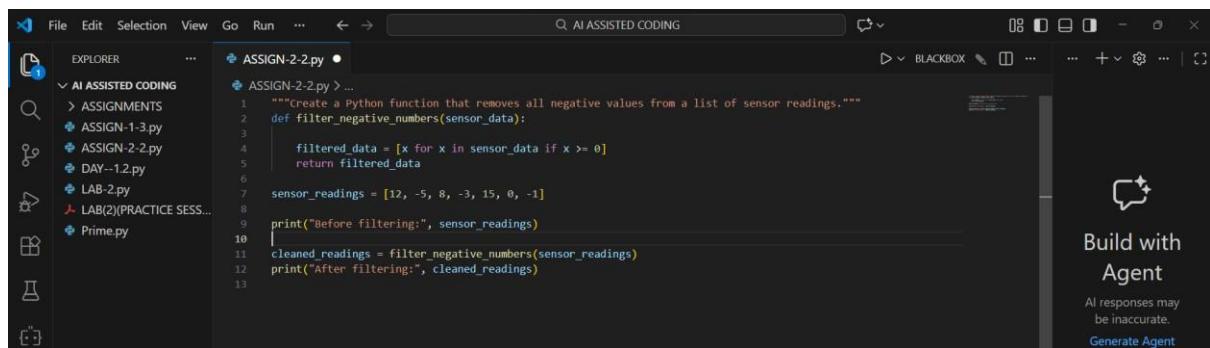
2403A51L31

B-52

ASSIGNMENT -2.2

Task 1: Cleaning Sensor Data

PROMPT: Create a Python function that removes all negative values from a list of sensor readings.



A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with several Python files. The main editor window contains the following code:

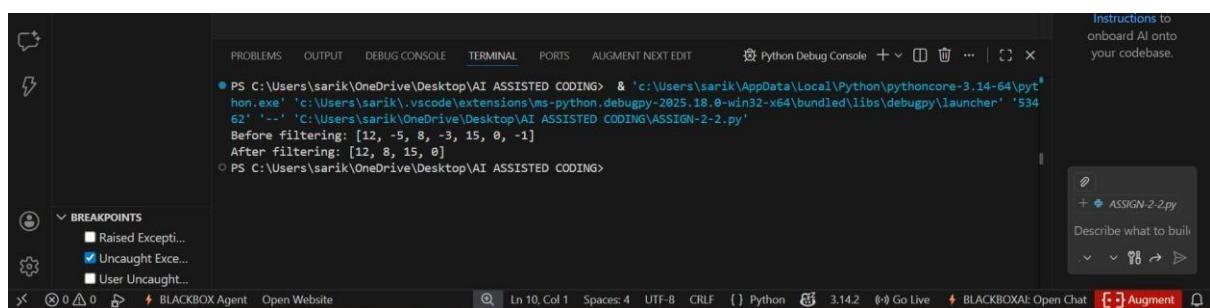
```
File Edit Selection View Go Run ... ⏪ ⏴ AI ASSISTED CODING ASSIGN-2-2.py ...
AI ASSISTED CODING
ASSIGNMENTS
ASSIGN-1-3.py
ASSIGN-2-2.py
DAY-1.2.py
LAB-2.py
LAB(2)(PRACTICE SESSIONS)
Prime.py

# Create a Python function that removes all negative values from a list of sensor readings.
def filter_negative_numbers(sensor_data):
    filtered_data = [x for x in sensor_data if x >= 0]
    return filtered_data

sensor_readings = [12, -5, 8, -3, 15, 0, -1]
print("Before filtering:", sensor_readings)
cleaned_readings = filter_negative_numbers(sensor_readings)
print("After filtering:", cleaned_readings)

12 8 15 0
```

OUTPUT:



A screenshot of the Visual Studio Code interface showing the terminal output. The terminal shows the execution of the Python script and its output:

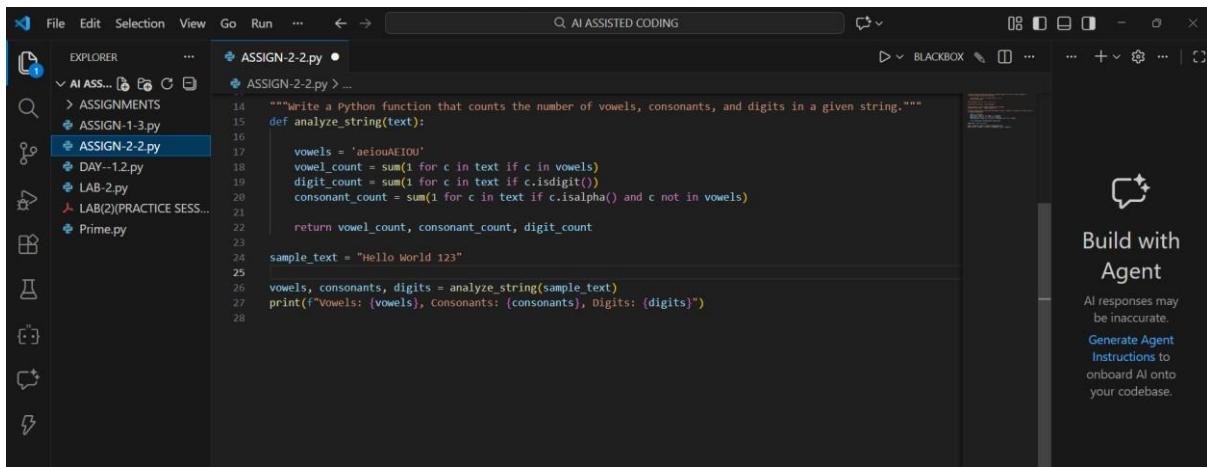
```
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING> & 'c:\Users\sarik\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\sarik\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53462' '--' 'C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING\ASSIGN-2-2.py'
Before filtering: [12, -5, 8, -3, 15, 0, -1]
After filtering: [12, 8, 15, 0]
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING>
```

EXPLANATION:

This function removes invalid negative sensor values using list comprehension. Only values greater than or equal to zero are retained, ensuring clean IoT sensor data.

Task 2: String Character Analysis

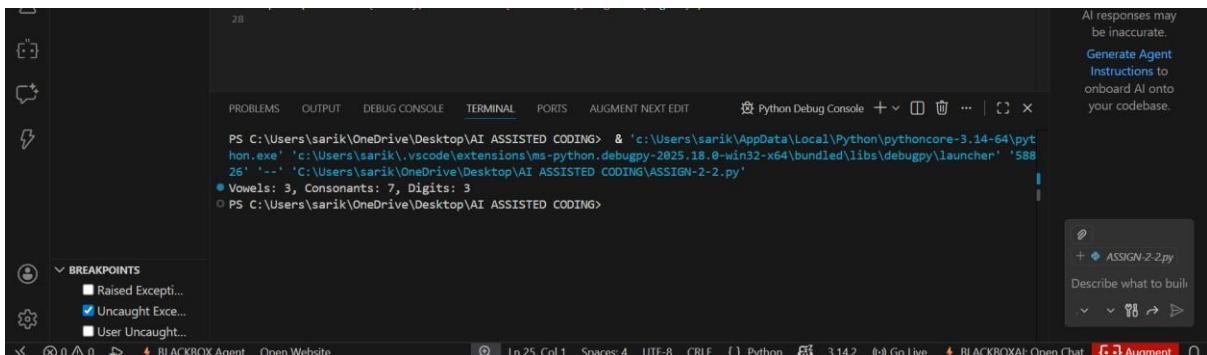
PROMPT: Write a Python function that counts the number of vowels, consonants, and digits in a given string.



The screenshot shows a dark-themed VS Code interface. In the Explorer sidebar, there are several projects and files listed under 'AI ASS...'. The file 'ASSIGN-2-2.py' is selected and shown in the main editor area. The code defines a function 'analyze_string' that takes a string 'text' and returns three counts: 'vowels', 'consonants', and 'digits'. It uses Python's built-in string methods like 'isalpha()' and 'isdigit()' to classify characters. A sample text 'Hello World 123' is passed to the function, and its output is printed to the terminal. A right-hand sidebar titled 'Build with Agent' provides instructions for using AI integration.

```
14 """Write a Python function that counts the number of vowels, consonants, and digits in a given string."""
15 def analyze_string(text):
16     vowels = 'aeiouAEIOU'
17     vowel_count = sum(1 for c in text if c in vowels)
18     digit_count = sum(1 for c in text if c.isdigit())
19     consonant_count = sum(1 for c in text if c.isalpha() and c not in vowels)
20
21     return vowel_count, consonant_count, digit_count
22
23
24 sample_text = "Hello World 123"
25
26 vowels, consonants, digits = analyze_string(sample_text)
27 print(f"Vowels: {vowels}, Consonants: {consonants}, Digits: {digits}")
```

OUTPUT:



The screenshot shows the terminal tab in VS Code displaying the execution of the 'ASSIGN-2-2.py' script. The terminal output shows the counts for the sample text 'Hello World 123': Vowels: 3, Consonants: 7, Digits: 3. The terminal also shows the command used to run the script and the current working directory.

```
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING> & 'c:\Users\sarik\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\sarik\.vscode\extensions\ms-python.debugpy-2025.8.0-win32-x64\bundled\libs\debugpy\launcher' '58826' '--' 'C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING\ASSIGN-2-2.py'
● Vowels: 3, Consonants: 7, Digits: 3
○ PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING>
```

EXPLANATION:

The function iterates through each character and classifies it as a vowel, consonant, or digit.

Python string methods like `isalpha()` and `isdigit()` improve accuracy and readability.

Task 3: Palindrome Check – Tool Comparison

Gemini Prompt: Write a Python function to check if a string is a palindrome. Ignore spaces and capitalization.

The screenshot shows the Visual Studio Code interface with the 'AI ASSISTED CODING' extension installed. The code editor displays a Python script named `ASSIGN-2-2.py`. The script includes functions for analyzing strings and checking for palindromes. A sidebar on the left provides options for running and debugging, and another sidebar on the right offers AI integration features like 'Build with Agent'.

```
File Edit Selection View Go Run ... ← → Q AI ASSISTED CODING ⚡ BLACKBOX ... + × ⚙ ... RUN AND DEBUG ... RUN Run and Debug To customize Run and Debug create a launch.json file. Debug using a terminal command or in an interactive chat. #Gemini Prompt: "Write a Python function to check if a string is a palindrome. ignore spaces and capitalization." def is_palindrome_gemini(s): s = s.replace(" ", "").lower() return s == s[::-1] print(is_palindrome_gemini("Racecar")) # True
```

OUTPUT:

The screenshot shows the VS Code interface with the Python extension running. The code editor displays a script with a palindrome check function. The terminal shows the command to run the script and its output, which is 'True'. The 'Breakpoints' sidebar shows three breakpoint types: 'Raised Excepti...', 'Uncaught Excepti...', and 'User Uncaught...'. The status bar at the bottom indicates the file is 'In 36 Col 40 Spaces: 4 UTE-8 CRLF'.

```
34     return s == s[::-1]
35
36 print(is_palindrome_gemini("Racecar"))
37
```

CALL STACK Running
Pyth... RUNNING

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AUGMENT NEXT EDIT Python Debug Console

PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING> & 'c:\Users\sarik\AppData\Local\Python\pythoncore-3.14-64\python.exe' 'c:\Users\sarik\.vscode\extensions\ms-python.python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '58783' '--' 'C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING\ASSIGN-2-2.py'
True

Breakpoints

- Raised Excepti...
- Uncaught Excepti...
- User Uncaught...

BLACKBOX Agent Open Website In 36 Col 40 Spaces: 4 UTE-8 CRLF Python 3.14.2 Go Live BLACKBOX: Open Chat Argument

Copilot Prompt: Write a Python function to check palindrome. Consider only letters and ignore case.

The screenshot shows the Visual Studio Code interface with the following details:

- File**, **Edit**, **Selection**, **View**, **Go**, **Run**, **...** menu bar.
- Toolbar icons: RUN AND DEBUG, RUN, Run and Debug (highlighted), Find, Cut/Copy/Paste, Select All, Undo/Redo.
- Left sidebar: RUN AND DEBUG, RUN, Run and Debug (highlighted), To customize Run and Debug create a launch.json file, Debug using a terminal command or in an interactive chat.
- Central area: A code editor with the file **ASSIGN-2-2.py** open. The code defines a function `is_palindrome_copilot` that checks if a string is a palindrome by ignoring case and non-alphanumeric characters. It includes a docstring and a test call to the function with the argument "Racecar".
- Right sidebar: BLACKBOX, Build with Agent (highlighted), AI Assistant icon, AI responses may be inaccurate, Generate Agent Instructions to onboard AI onto your codebase.

OUTPUT:

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The terminal tab is active, displaying the command PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING & C:/Users/sarik/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/sarik/OneDrive/Desktop/AI ASSISTED CODING/ASSIGN-2-2.py". The output pane shows the result: True. The code editor pane contains the Python code for the assignment. The sidebar on the left shows the file tree and includes a 'BREAKPOINTS' section with three items: 'Raised Excepti...', 'Uncaught Exce...', and 'User Uncaught...'. A small icon for 'BLACKBOX Agent' is also present. On the right, there is an 'Agent' panel with the text 'AI responses may be inaccurate.' and a button 'Generate Agent Instructions to onboard AI onto your codebase.' Below this is a tooltip for 'ASSIGN-2-2.py' with the text 'Describe what to build'.

Comparison Table:

Feature	Gemini	Copilot
Clarity	Simple, minimal code	Slightly longer, more robust
Handling spaces/case	Ignores spaces, converts to lowercase	Ignores spaces and punctuation, lowercase
Readability	Very clear	Clear, slightly more detailed
Efficiency	Uses string slicing	Uses string comprehension

EXPLANATION:

Gemini provides concise and easy-to-read logic, making it beginnerfriendly. Copilot generates more robust code that handles punctuation and special characters.

Task 4: Code Explanation Using AI Step 1 –

Code Snippet:

The screenshot shows a code editor interface with a dark theme. On the left, there's a sidebar with icons for file operations like Open, Save, and Run, along with a search bar and a 'Run and Debug' button. The main area displays a Python script named 'ASSIGN-2-2.py'. The code contains two lines of Python code:

```
49     ##Step 1 - Code Snippet(Code Explanation):  
50     def is_palindrome(text):  
51         text = text.replace(" ", "").lower() # Remove spaces and lowercase  
52         return text == text[::-1] # Compare string with its reverse
```

To the right of the code, there's a 'Build with Agent' panel. It features a speech bubble icon and the text 'Build with Agent'. Below it, a note says 'AI responses may be inaccurate.' and 'Generate Agent Instructions to onboard AI onto your codebase.'

Step 2 – AI Explanation:

1. `text.replace(" ", "").lower()` → Removes spaces and converts letters to lowercase.
2. `text == text[::-1]` → Checks if the string is equal to its reverse.

EXPLANATION:

The function normalizes the string to avoid case and space mismatches. It then compares the string with its reverse to verify palindrome logic.