SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE	**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

Program Name: B. Tech	**Assignment Type: Lab**	**Academic Year:**2025-2026

Course Coordinator Name	Venkataramana Veeramsetty
Instructor(s) Name	Dr. V. Venkataramana (Co-Ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij Kishor Tiwari Dr.J.Ravichander Dr. Mohammand Ali Shaik Dr. Anirodh Kumar Mr. S.Naresh Kumar Dr. RAJESH VELPULA Mr. Kundhan Kumar Ms. Ch.Rajitha Mr. M Prakash Mr. B.Raju Intern 1 (Dharma teja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS_2 (Mounika)

Course Code	24CS002PC215	**Course Title**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week2 - Monday	**Time(s)**	
Duration	2 Hours	**Applicable to Batches**	

Assignment Number:4.1(Present assignment number)/**24**(Total number of assignments)

Q.No.	**Question**	***Expected Time to complete***
1	Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques **Lab Objectives:** • To explore and apply different levels of prompt examples in AI-assisted code generation.	Week2 - Monday

- To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality.
- To evaluate the impact of context richness and example quantity on AI performance.
- To build awareness of prompt strategy effectiveness for different problem types.

Lab Outcomes (LOs):

After completing this lab, students will be able to:

- Use zero-shot prompting to instruct AI with minimal context.
- Use one-shot prompting with a single example to guide AI code generation.
- Apply few-shot prompting using multiple examples to improve AI responses.
- Compare AI outputs across the three prompting strategies.

Task #1 – Zero-Shot Prompting with Conditional Validation

Objective

Use zero-shot prompting to instruct an AI tool to generate a function that validates an Indian mobile number.

Requirements

- The function must ensure the mobile number:
  - Starts with 6, 7, 8, or 9
  - Contains exactly 10 digits

Expected Output

- A valid Python function that performs all required validations without using any input-output examples in the prompt.

Task #2 – One-Shot Prompting with Edge Case Handling

Objective

Use one-shot prompting to generate a Python function that calculates the factorial of a number.

Requirements

- Provide one sample input-output pair in the prompt to guide the AI.
- The function should handle:
  - 0! correctly
  - Negative input by returning an appropriate message

Expected Output

- A Python function with correct factorial logic and edge case handling, generated from a single example.

Task #3 – Few-Shot Prompting for Nested Dictionary Extraction

Objective

Use few-shot prompting (2–3 examples) to instruct the AI to create a function that parses a nested dictionary representing student information.

Requirements

- The function should extract and return:
  - Full Name
  - Branch
  - SGPA

Expected Output

- A reusable Python function that correctly navigates and extracts values from nested dictionaries based on the provided examples.

Task #4 – Comparing Prompting Styles for File Analysis

Objective

Experiment with zero-shot, one-shot, and few-shot prompting to generate functions for CSV file analysis.

Requirements

- Each generated function should:
  - Read a .csv file
  - Return the total number of rows
  - Count the number of empty rows
  - Count the number of words across the file

Expected Output

- Working Python functions for each prompting style, with a brief reflection comparing their accuracy, clarity, and efficiency.

Task #5 – Few-Shot Prompting for Text Processing and Word Frequency

Objective

Use few-shot prompting (with at least 3 examples) to generate a Python function that processes text and analyzes word frequency.

Requirements

The function must:

- Accept a paragraph as input
- Convert all text to lowercase
- Remove punctuation
- Return the most frequently used word

Expected Output

- A functional Python script that performs text cleaning, tokenization, and returns the most common word using only the examples provided in the prompt

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

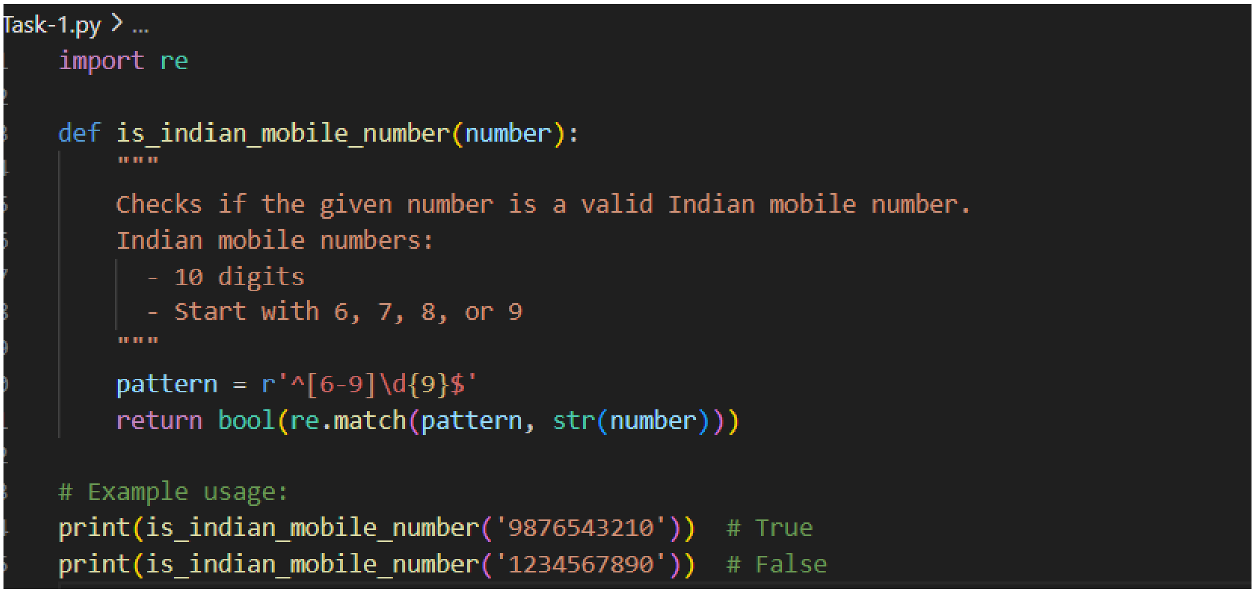
Criteria	**Max Marks**
Zero Shot (Task #1)	0.5
One Shot (Task#2)	0.5
Few Shot (Task#3, Task#4 & Task #5)	1.5
Total	**2.5 Marks**

GITHUB COPILOT:

TASK-1:

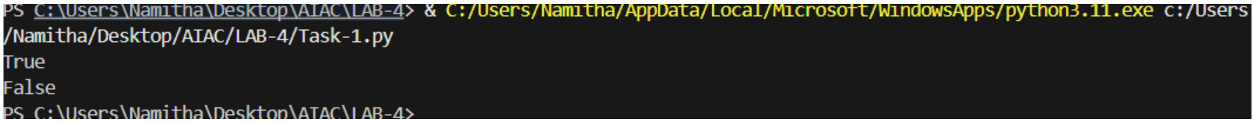
PROMPT: write a python function that checks whether given number is Indian mobile number or not.

Code:



```python
import re

def is_indian_mobile_number(number):
    """
    Checks if the given number is a valid Indian mobile number.
    Indian mobile numbers:
      - 10 digits
      - Start with 6, 7, 8, or 9
    """
    pattern = r'^[6-9]\d{9}$'
    return bool(re.match(pattern, str(number)))

# Example usage:
print(is_indian_mobile_number('9876543210'))  # True
print(is_indian_mobile_number('1234567890'))  # False
```

Output:

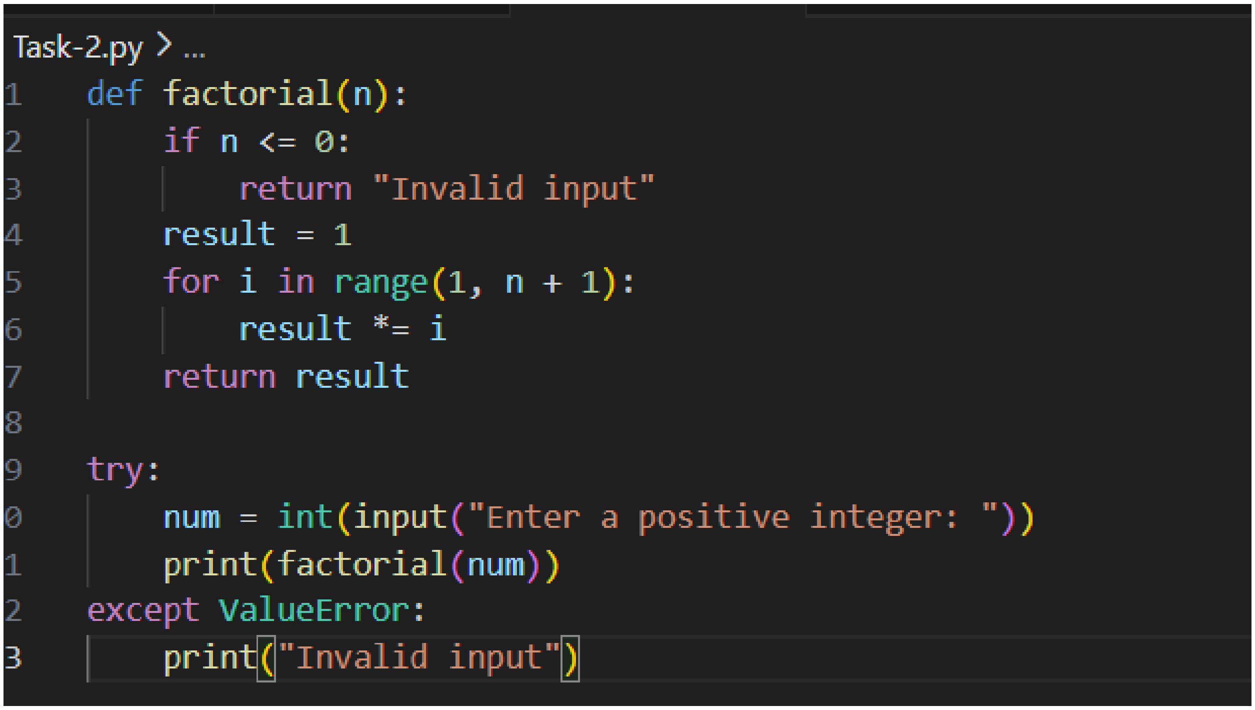


```
PS C:\Users\Namitha\Desktop\AIAC\LAB-4> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users
/Namitha/Desktop/AIAC/LAB-4/Task-1.py
True
False
PS C:\Users\Namitha\Desktop\AIAC\LAB-4>
```

TASK-2:

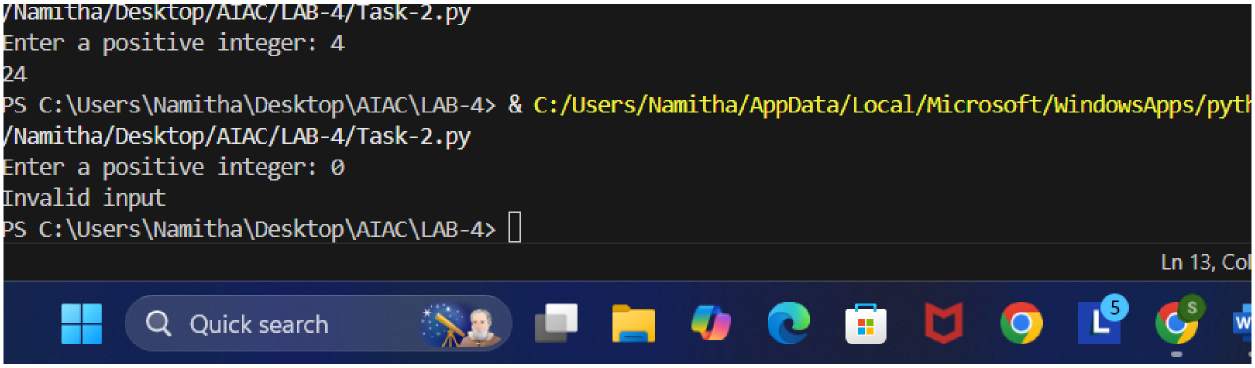
PROMPT: write a python function that calculates factorial of given number and return invalid input if given number is zero or negative.Example if input is 5 ,output should be 120. Take input from user.

Code:



```python
def factorial(n):
    if n <= 0:
        return "Invalid input"
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result

try:
    num = int(input("Enter a positive integer: "))
    print(factorial(num))
except ValueError:
    print("Invalid input")
```

Output:

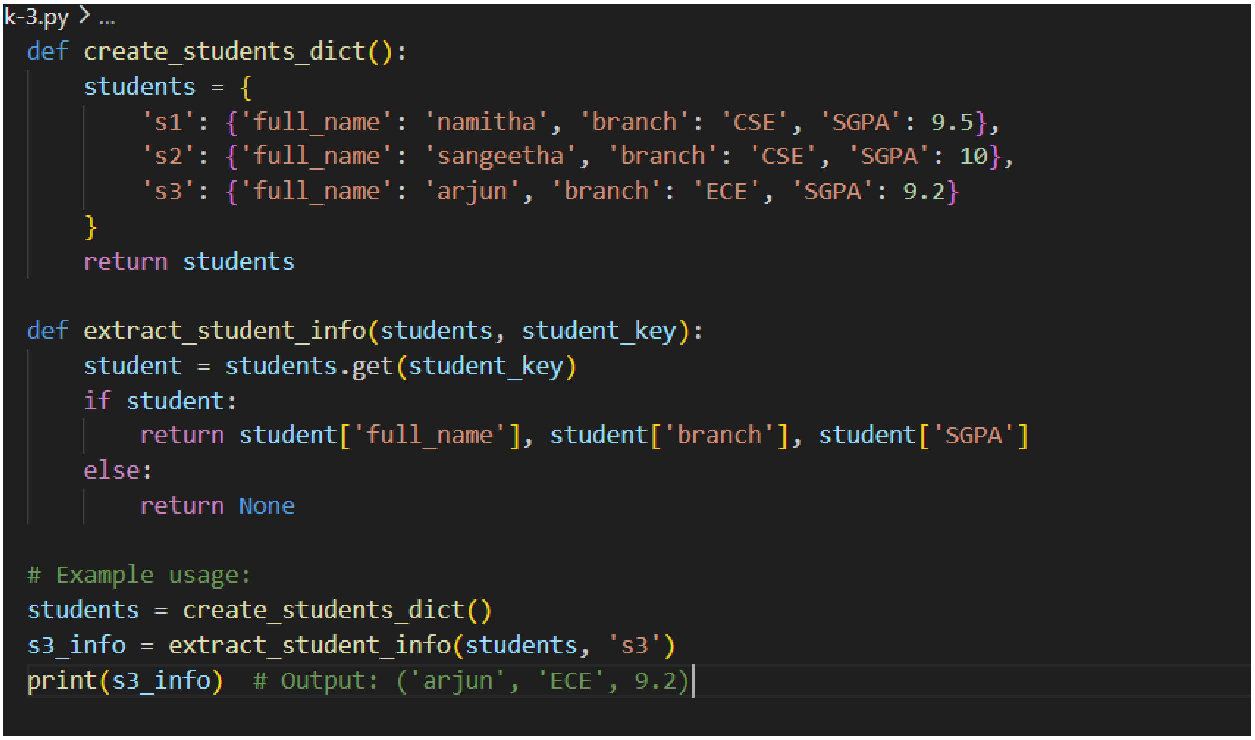


```
/Namitha/Desktop/AIAC/LAB-4/Task-2.py
Enter a positive integer: 4
24
PS C:\Users\Namitha\Desktop\AIAC\LAB-4> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/pyth
/Namitha/Desktop/AIAC/LAB-4/Task-2.py
Enter a positive integer: 0
Invalid input
PS C:\Users\Namitha\Desktop\AIAC\LAB-4>
```

TASK-3:

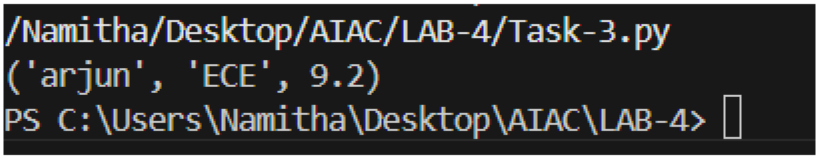
PROMPT: write a python function that creates nested dictionary representing student information .that function should extract and return full-name,branch,SGPA. example s1=full_name:namitha,branch:CSE,SGPA:9.5,s2=full_name:sangeetha,branch:CSE,SGPA:10. now extract and return for s3.

CODE:



```python
def create_students_dict():
    students = {
        's1': {'full_name': 'namitha', 'branch': 'CSE', 'SGPA': 9.5},
        's2': {'full_name': 'sangeetha', 'branch': 'CSE', 'SGPA': 10},
        's3': {'full_name': 'arjun', 'branch': 'ECE', 'SGPA': 9.2}
    }
    return students

def extract_student_info(students, student_key):
    student = students.get(student_key)
    if student:
        return student['full_name'], student['branch'], student['SGPA']
    else:
        return None

# Example usage:
students = create_students_dict()
s3_info = extract_student_info(students, 's3')
print(s3_info)  # Output: ('arjun', 'ECE', 9.2)
```

OUTPUT:



```
/Namitha/Desktop/AIAC/LAB-4/Task-3.py
('arjun', 'ECE', 9.2)
PS C:\Users\Namitha\Desktop\AIAC\LAB-4> 
```

TASK-5:

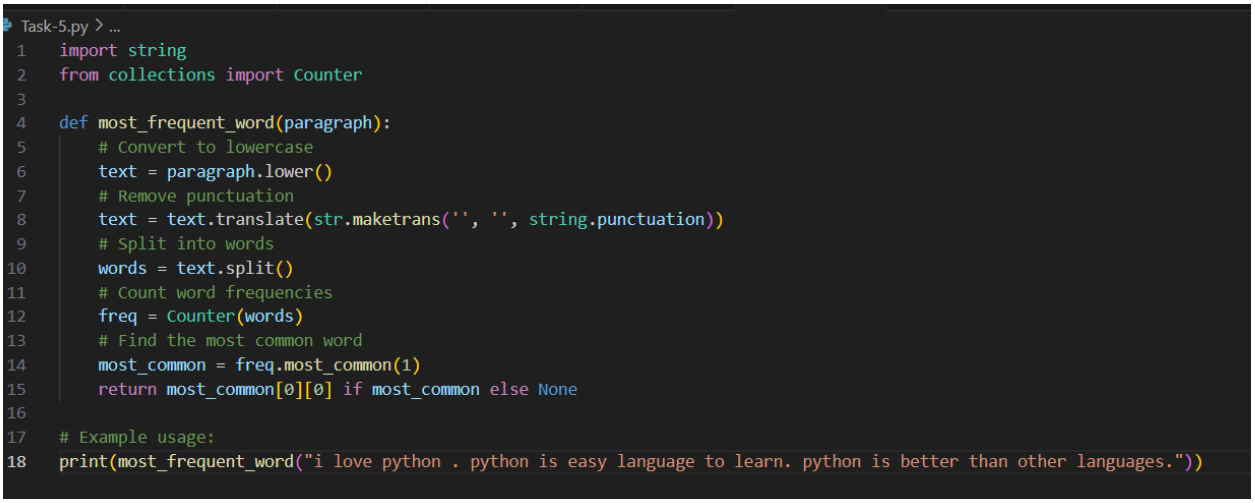
PROMPT: write a python function that process the text and analyze word frequency.
The function must:

- Accept a paragraph as input
- Convert all text to lowercase
- Remove punctuation
- Return the most frequently used word

example1text: her name is dora.dora is a good girl.dora always explores the world.dora cant see properly. here output should be dora.
example2 text: betty bought some better butter but it was bitter butter so betty add some more better

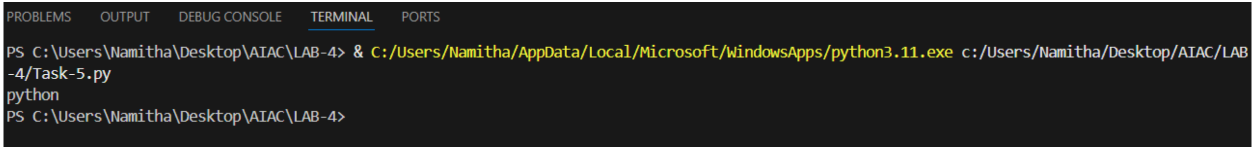
butter and made it bitter butter better. output must be butter.
example3 text: my name is nammu. my course in btech is cse . is this better? output should be is.now for example text4: i love python . python is easy language to learn. python is better than other languages. now what is the output?

Code:



```
import string
from collections import Counter

def most_frequent_word(paragraph):
    # Convert to lowercase
    text = paragraph.lower()
    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))
    # Split into words
    words = text.split()
    # Count word frequencies
    freq = Counter(words)
    # Find the most common word
    most_common = freq.most_common(1)
    return most_common[0][0] if most_common else None

# Example usage:
print(most_frequent_word("i love python . python is easy language to learn. python is better than other languages."))
```

Output:



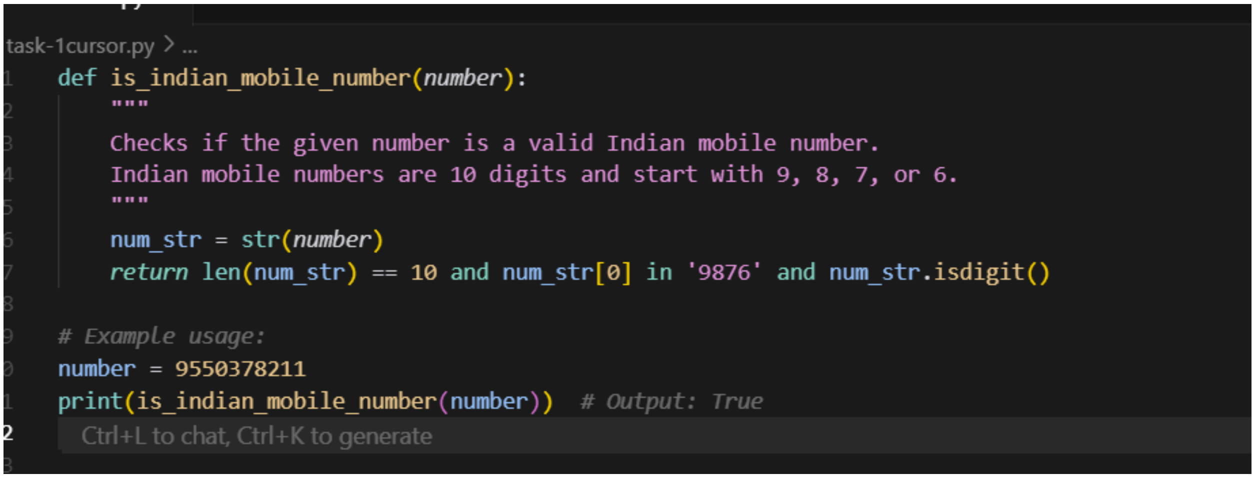
```
PS C:\Users\Namitha\Desktop\AIAC\LAB-4> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Namitha/Desktop/AIAC/LAB
-4/Task-5.py
python
PS C:\Users\Namitha\Desktop\AIAC\LAB-4>
```

CURSOR AI:

TASK-1:

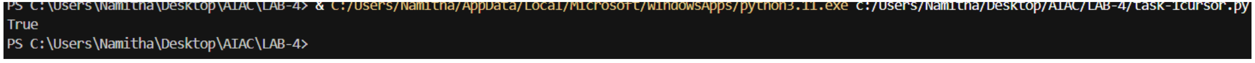
PROMPT:write a python function that checks whether given number is Indian mobile number or not. number is 9550378211.

Code:



```python
def is_indian_mobile_number(number):
    """
    Checks if the given number is a valid Indian mobile number.
    Indian mobile numbers are 10 digits and start with 9, 8, 7, or 6.
    """
    num_str = str(number)
    return len(num_str) == 10 and num_str[0] in '9876' and num_str.isdigit()

# Example usage:
number = 9550378211
print(is_indian_mobile_number(number))  # Output: True
```

Output:

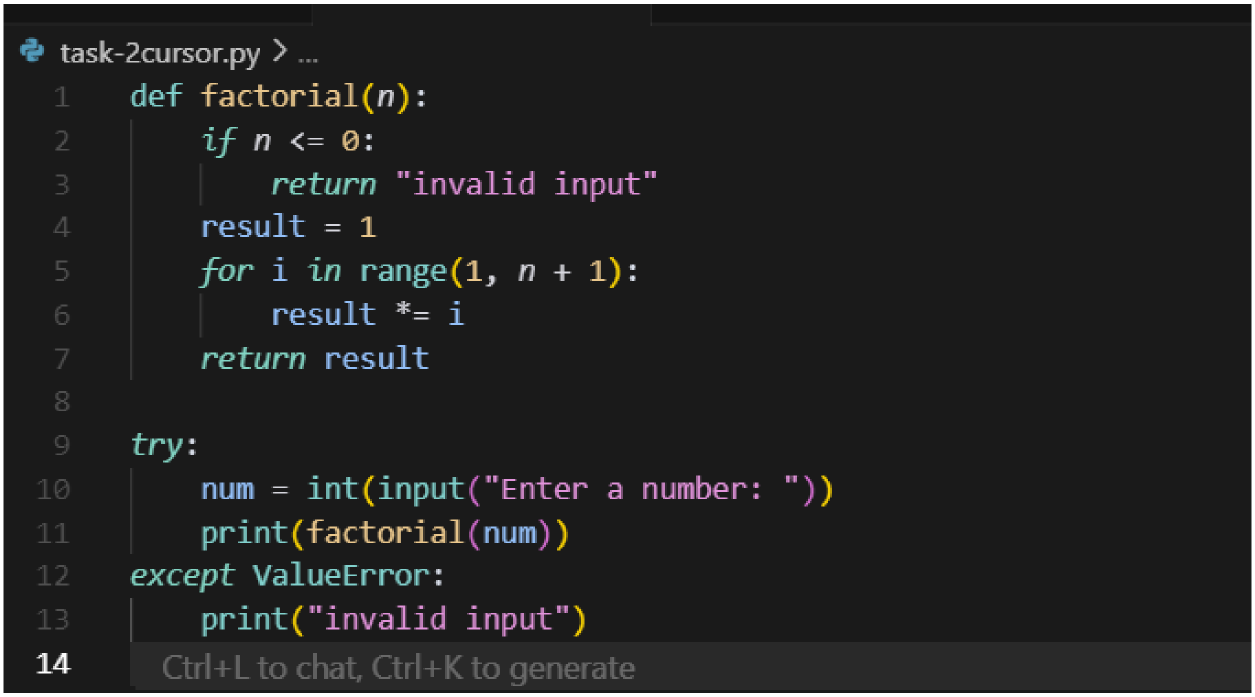


```
PS C:\Users\Namitha\Desktop\AIAC\LAB-4> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Namitha/Desktop/AIAC/LAB-4/task-1cursor.py
True
PS C:\Users\Namitha\Desktop\AIAC\LAB-4>
```

TASK-2:

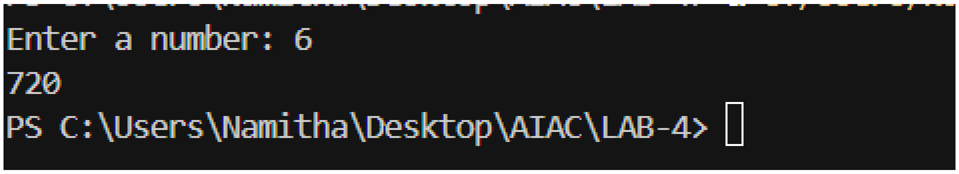
PROMPT: write a python function that calculates factorial of given number and return invalid input if given number is zero or negative.Example if input is 5 ,output should be 120. Take input from user.

Code:



```python
def factorial(n):
    if n <= 0:
        return "invalid input"
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result

try:
    num = int(input("Enter a number: "))
    print(factorial(num))
except ValueError:
    print("invalid input")
```

Output:

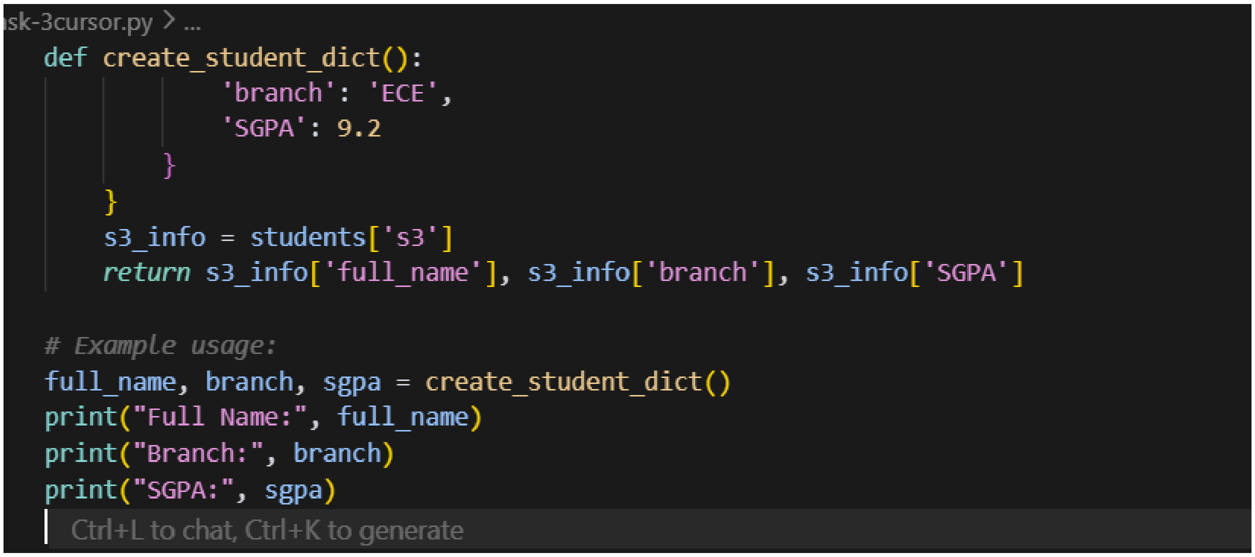


```
Enter a number: 6
720
PS C:\Users\Namitha\Desktop\AIAC\LAB-4>
```

TASK-3:

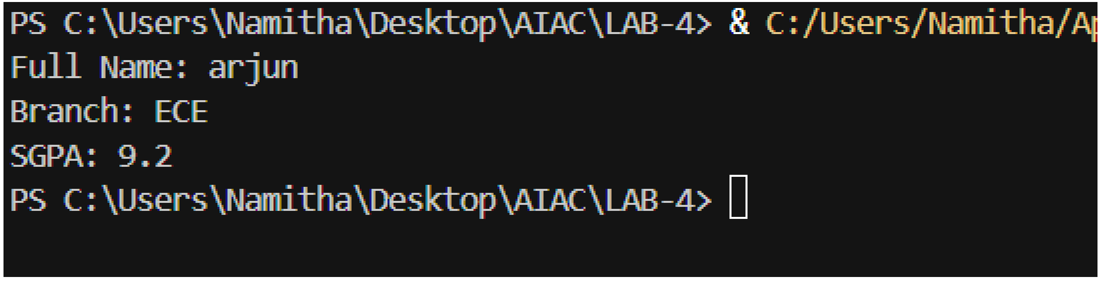
PROMPT: write a python function that creates nested dictionary representing student information .that function should extract and return full-name,branch,SGPA. example s1=full_name:namitha,branch:CSE,SGPA:9.5,s2=full_name:sangeetha,branch:CSE,SGPA:10. now extract and return for s3.

Code:



```
sk-3cursor.py > ...
def create_student_dict():
            'branch': 'ECE',
            'SGPA': 9.2
        }
    }
    s3_info = students['s3']
    return s3_info['full_name'], s3_info['branch'], s3_info['SGPA']

# Example usage:
full_name, branch, sgpa = create_student_dict()
print("Full Name:", full_name)
print("Branch:", branch)
print("SGPA:", sgpa)
```

Output:



```
PS C:\Users\Namitha\Desktop\AIAC\LAB-4> & C:/Users/Namitha/A
Full Name: arjun
Branch: ECE
SGPA: 9.2
PS C:\Users\Namitha\Desktop\AIAC\LAB-4>
```

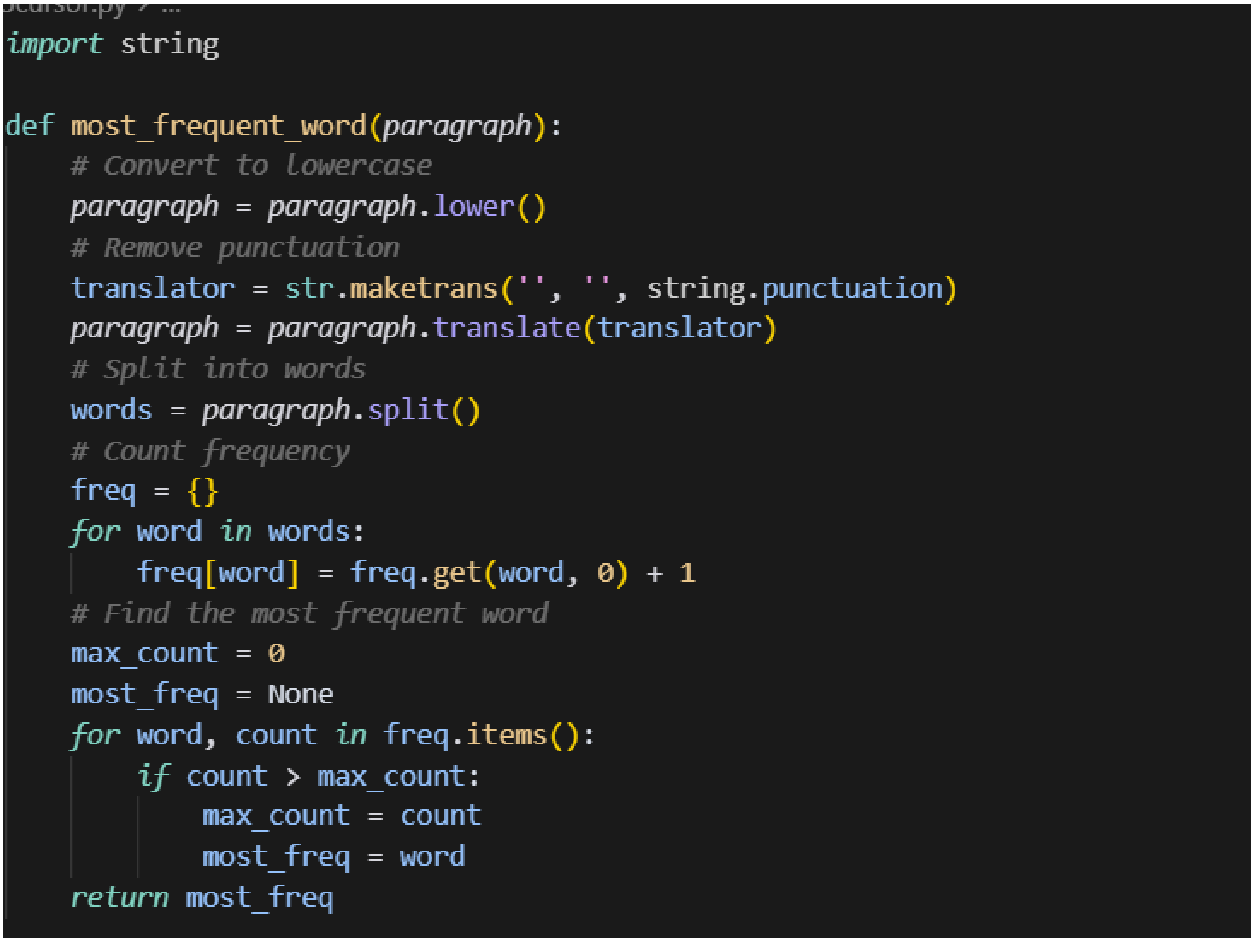
TASK-5:

PROMPT: write a python function that process the text and analyze word frequency.
The function must:

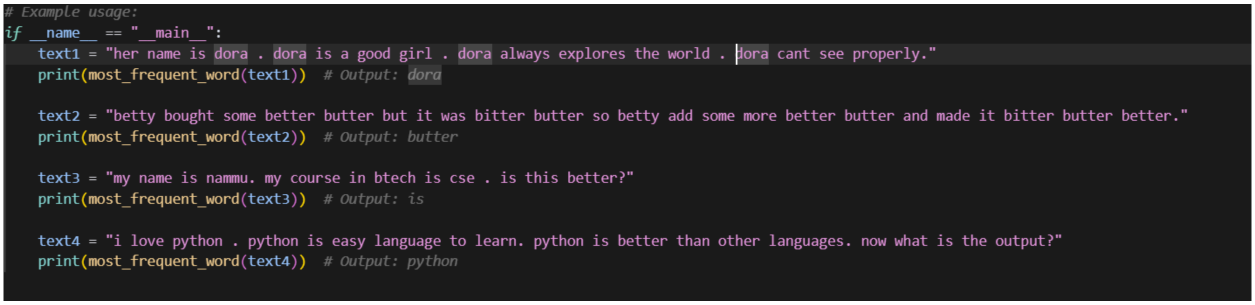
- Accept a paragraph as input
- Convert all text to lowercase
- Remove punctuation
- Return the most frequently used word

example1text: her name is dora.dora is a good girl.dora always explores the world.dora cant see properly. here output should be dora.
example2 text: betty bought some better butter but it was bitter butter so betty add some more better butter and made it bitter butter better. output must be butter.
example3 text: my name is nammu. my course in btech is cse . is this better? output should be is.now for example text4: i love python . python is easy language to learn. python is better than other languages. now what is the output?

Code:

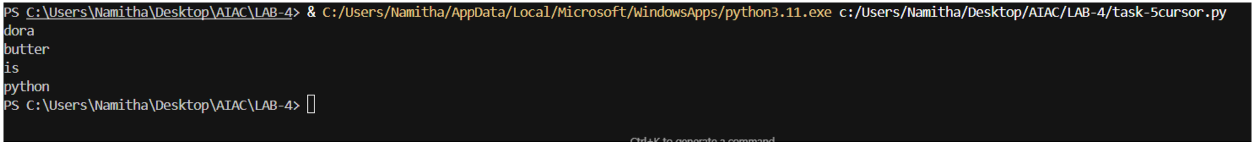


```python
import string

def most_frequent_word(paragraph):
    # Convert to lowercase
    paragraph = paragraph.lower()
    # Remove punctuation
    translator = str.maketrans('', '', string.punctuation)
    paragraph = paragraph.translate(translator)
    # Split into words
    words = paragraph.split()
    # Count frequency
    freq = {}
    for word in words:
        freq[word] = freq.get(word, 0) + 1
    # Find the most frequent word
    max_count = 0
    most_freq = None
    for word, count in freq.items():
        if count > max_count:
            max_count = count
            most_freq = word
    return most_freq
```



```python
# Example usage:
if __name__ == "__main__":
    text1 = "her name is dora . dora is a good girl . dora always explores the world . dora cant see properly."
    print(most_frequent_word(text1))  # Output: dora

    text2 = "betty bought some better butter but it was bitter butter so betty add some more better butter and made it bitter butter better."
    print(most_frequent_word(text2))  # Output: butter

    text3 = "my name is nammu. my course in btech is cse . is this better?"
    print(most_frequent_word(text3))  # Output: is

    text4 = "i love python . python is easy language to learn. python is better than other languages. now what is the output?"
    print(most_frequent_word(text4))  # Output: python
```

Output:



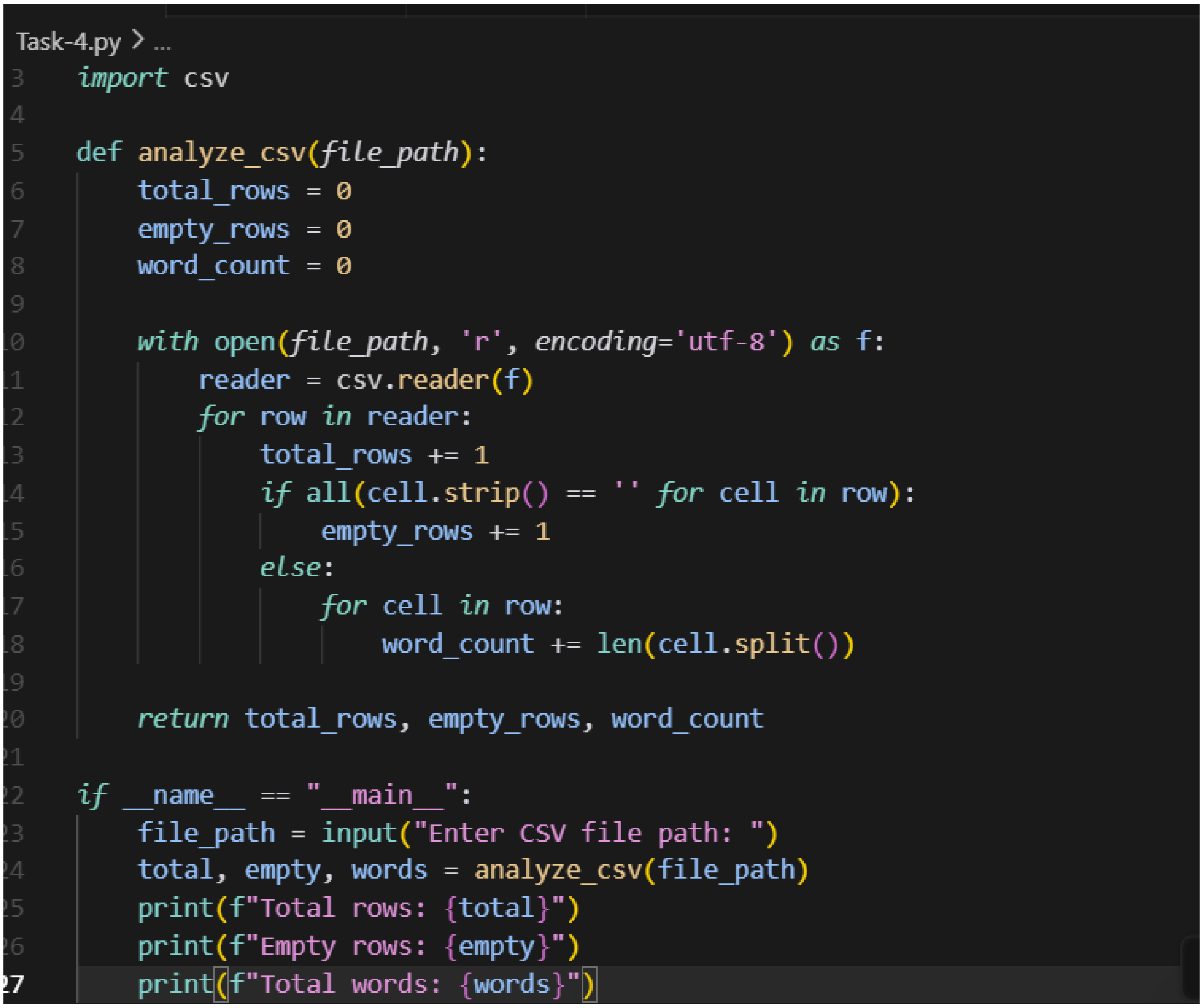
```
PS C:\Users\Namitha\Desktop\AIAC\LAB-4> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Namitha/Desktop/AIAC/LAB-4/task-5cursor.py
dora
butter
is
python
PS C:\Users\Namitha\Desktop\AIAC\LAB-4> 
```

TASK-4:

PROMPT1: : Write a Python function that reads a CSV file and returns:

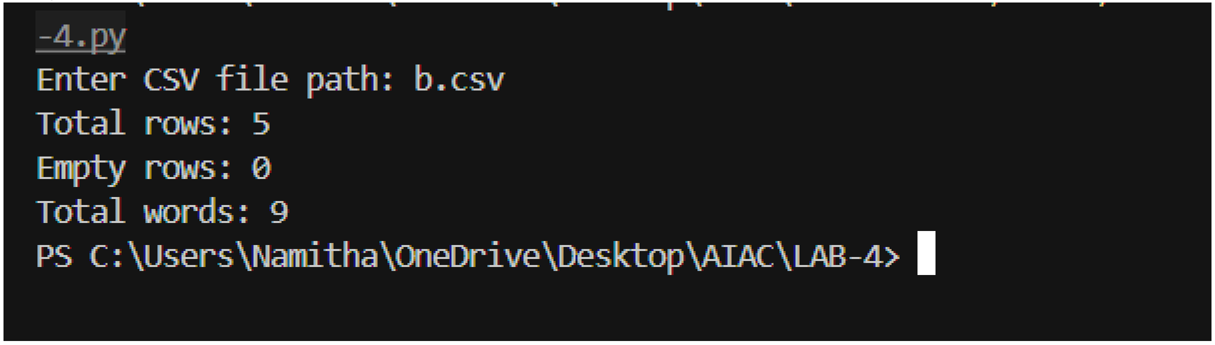
- The total number of rows
- The number of empty rows
- The total number of words across the file

Code:



```python
import csv

def analyze_csv(file_path):
    total_rows = 0
    empty_rows = 0
    word_count = 0

    with open(file_path, 'r', encoding='utf-8') as f:
        reader = csv.reader(f)
        for row in reader:
            total_rows += 1
            if all(cell.strip() == '' for cell in row):
                empty_rows += 1
            else:
                for cell in row:
                    word_count += len(cell.split())

    return total_rows, empty_rows, word_count

if __name__ == "__main__":
    file_path = input("Enter CSV file path: ")
    total, empty, words = analyze_csv(file_path)
    print(f"Total rows: {total}")
    print(f"Empty rows: {empty}")
    print(f"Total words: {words}")
```

Output:

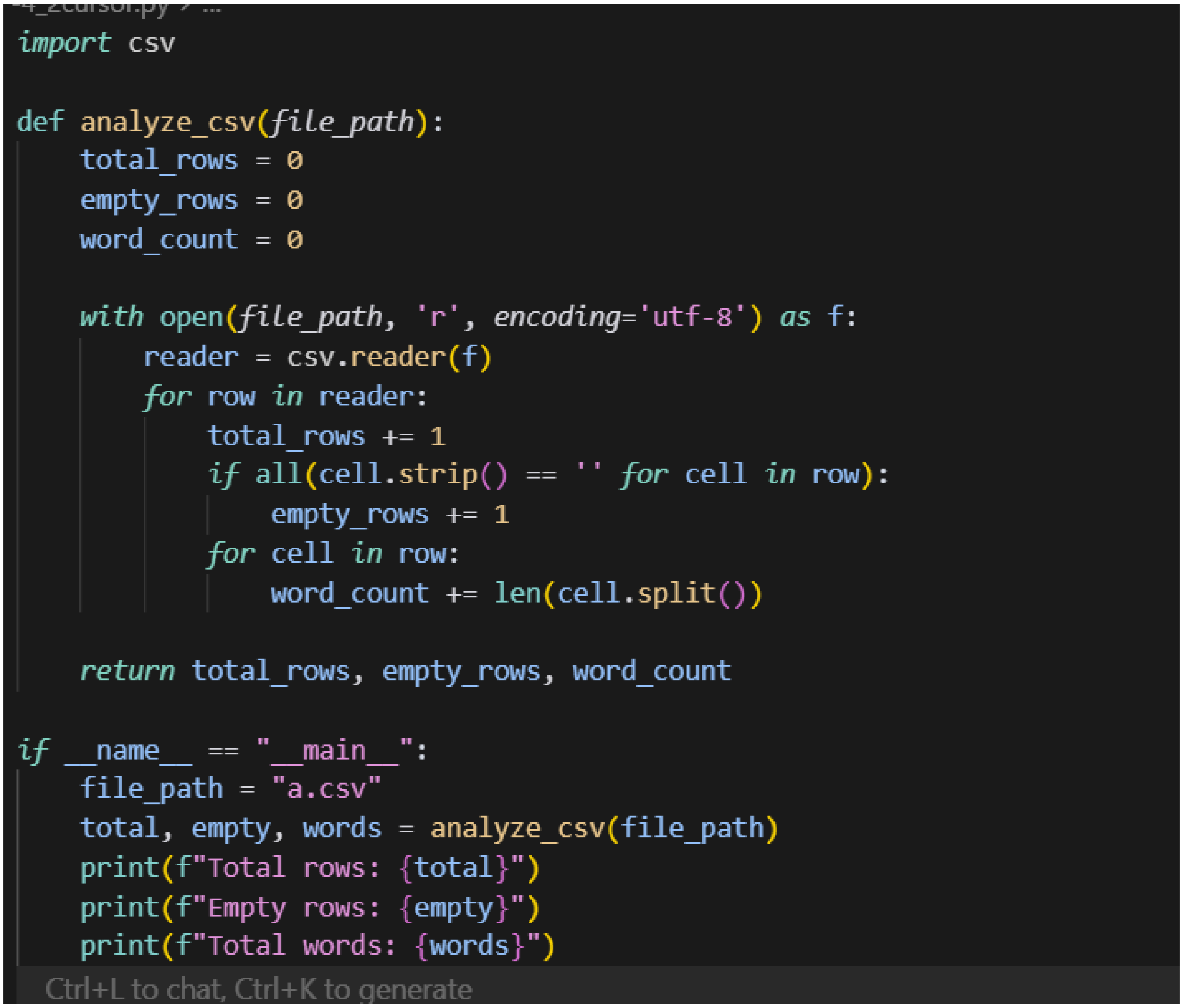


```
-4.py
Enter CSV file path: b.csv
Total rows: 5
Empty rows: 0
Total words: 9
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\LAB-4>
```

PROMPT-2: Now, write a Python function that reads a CSV file and returns:

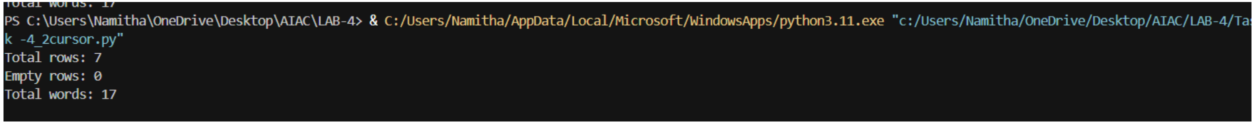
- The total number of rows
- The number of empty rows
- The total number of words across the file
- EXAMPLE: in b.csv file we have 5 rows,empty rows=0,total rows=9.
- Find in a.csv

Code:



```python
import csv

def analyze_csv(file_path):
    total_rows = 0
    empty_rows = 0
    word_count = 0

    with open(file_path, 'r', encoding='utf-8') as f:
        reader = csv.reader(f)
        for row in reader:
            total_rows += 1
            if all(cell.strip() == '' for cell in row):
                empty_rows += 1
            for cell in row:
                word_count += len(cell.split())

    return total_rows, empty_rows, word_count

if __name__ == "__main__":
    file_path = "a.csv"
    total, empty, words = analyze_csv(file_path)
    print(f"Total rows: {total}")
    print(f"Empty rows: {empty}")
    print(f"Total words: {words}")
```

Output:



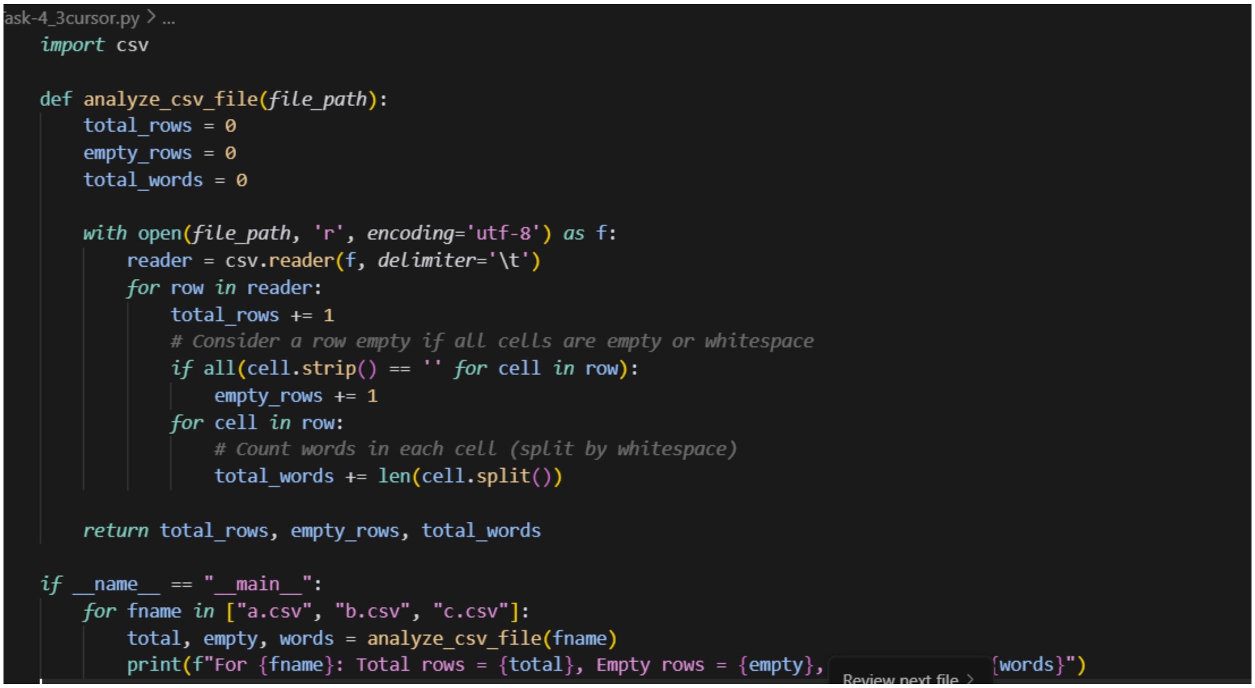
```
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\LAB-4> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/Namitha/OneDrive/Desktop/AIAC/LAB-4/Ta
k -4_2cursor.py"
Total rows: 7
Empty rows: 0
Total words: 17
```

PROMPT3: Now, write a Python function that reads a CSV file and returns:

- The total number of rows
- The number of empty rows
- The total number of words across the file
- EXAMPLE1: in b.csv we have 5 rows,empty rows=0,total words=9.
- example2: in a.csv file we have 7 rows,0 empty rows,17 total words

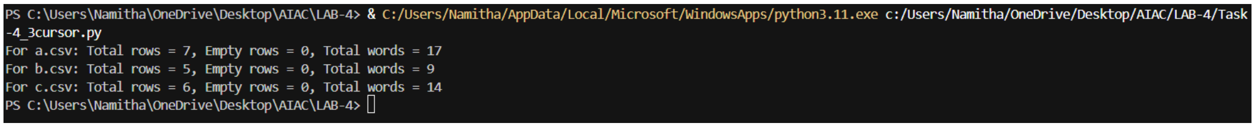
in similar way find for c.csv

CODE:



```
import csv

def analyze_csv_file(file_path):
    total_rows = 0
    empty_rows = 0
    total_words = 0

    with open(file_path, 'r', encoding='utf-8') as f:
        reader = csv.reader(f, delimiter='\t')
        for row in reader:
            total_rows += 1
            # Consider a row empty if all cells are empty or whitespace
            if all(cell.strip() == '' for cell in row):
                empty_rows += 1
            for cell in row:
                # Count words in each cell (split by whitespace)
                total_words += len(cell.split())

    return total_rows, empty_rows, total_words

if __name__ == "__main__":
    for fname in ["a.csv", "b.csv", "c.csv"]:
        total, empty, words = analyze_csv_file(fname)
        print(f"For {fname}: Total rows = {total}, Empty rows = {empty},               {words}")
```

OUTPUT:



```
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\LAB-4> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Namitha/OneDrive/Desktop/AIAC/LAB-4/Task
-4_3cursor.py
For a.csv: Total rows = 7, Empty rows = 0, Total words = 17
For b.csv: Total rows = 5, Empty rows = 0, Total words = 9
For c.csv: Total rows = 6, Empty rows = 0, Total words = 14
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\LAB-4> 
```