LAB ASSIGNMENT

Name: P. Reshmitha Reddy Roll No: 2403A510A7

Batch: 05 Course: AI Assisted coding

Q1. Zero-shot Classification[5M]

• Task 1: Write a zero-shot prompt to classify sentiment without any examples.

• Task 2: Create a scenario where an AI assistant needs to help a student solve math problems.

Write two prompts: one without context and one with detailed context (e.g., grade level, topic,

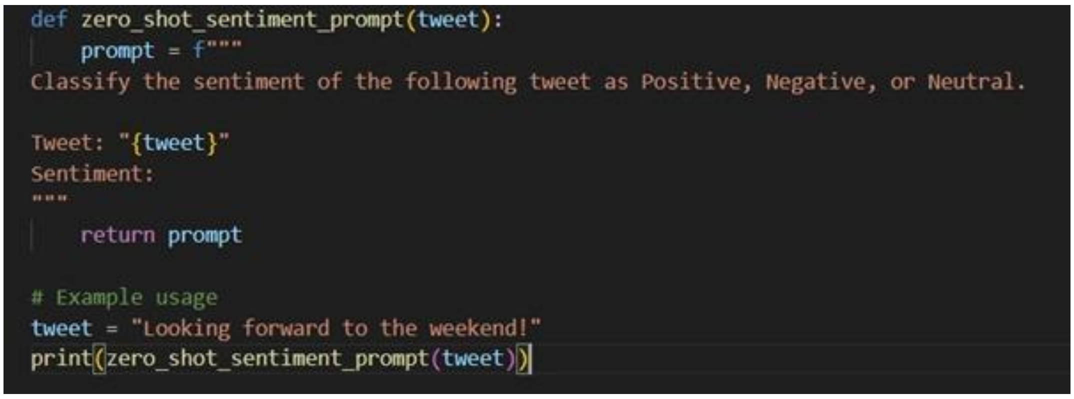
difficulty)

• Task 1: Write a zero-shot prompt to classify sentiment without any examples.

Prompt:

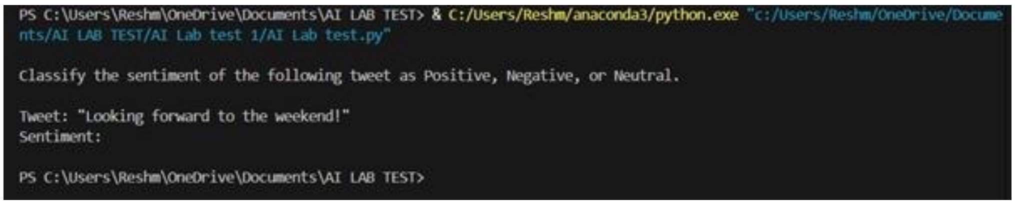
#Task 1: Write a zero-shot prompt to classify sentiment without any examples.

Code:



```python
def zero_shot_sentiment_prompt(tweet):
    prompt = f"""
Classify the sentiment of the following tweet as Positive, Negative, or Neutral.

Tweet: "{tweet}"
Sentiment:
"""
    return prompt

# Example usage
tweet = "Looking forward to the weekend!"
print(zero_shot_sentiment_prompt(tweet))
```

Output:



```
PS C:\Users\Reshm\OneDrive\Documents\AI LAB TEST> & C:/Users/Reshm/anaconda3/python.exe "c:/Users/Reshm/OneDrive/Docume
nts/AI LAB TEST/AI Lab test 1/AI Lab test.py"

Classify the sentiment of the following tweet as Positive, Negative, or Neutral.

Tweet: "Looking forward to the weekend!"
Sentiment:

PS C:\Users\Reshm\OneDrive\Documents\AI LAB TEST>
```

Observation:

The provided code defines a function that generates a zero-shot prompt for sentiment classification of tweets. The prompt asks to classify the sentiment of a given tweet as "Positive," "Negative," or "Neutral" without providing any examples. The function is demonstrated with the tweet "Looking forward to the weekend!" and prints the resulting prompt. This setup is suitable for use with large language models or APIs that support zero-shot classification.

Task 2: Create a scenario where an AI assistant needs to help a student solve math problems.

Write two prompts: one without context and one with detailed context (e.g., grade level, topic,

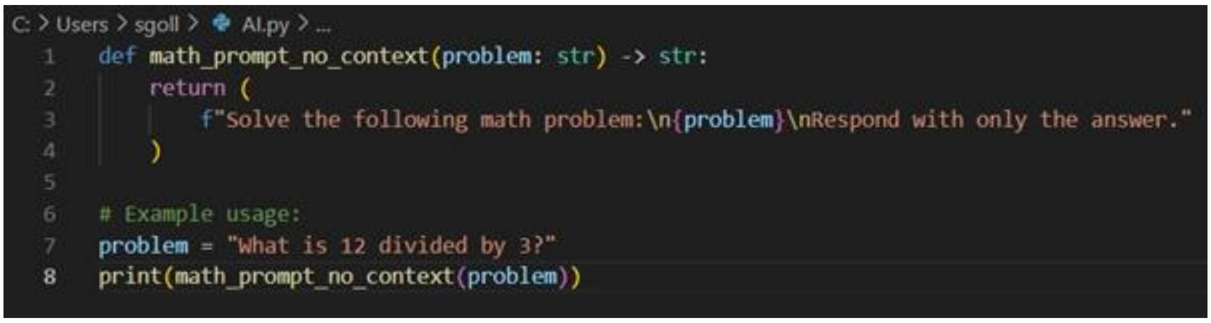
difficulty).

Prompt:

# Create a scenario where an AI assistant needs to help a student solve math problems.

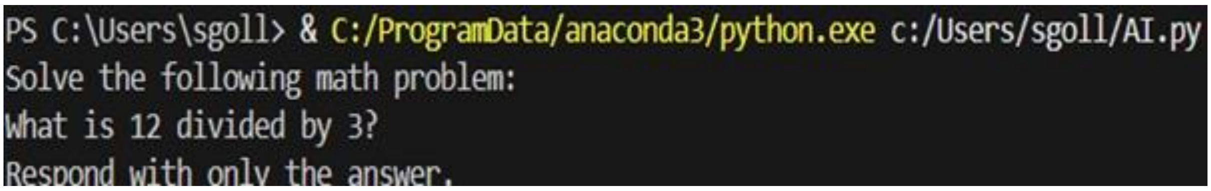
give one code without content

Code:



```
C: > Users > sgoll > AI.py > ...
def math_prompt_no_context(problem: str) -> str:
    return (
        f"Solve the following math problem:\n{problem}\nRespond with only the answer."
    )

# Example usage:
problem = "What is 12 divided by 3?"
print(math_prompt_no_context(problem))
```

Output:



```
PS C:\Users\sgoll> & C:/ProgramData/anaconda3/python.exe c:/Users/sgoll/AI.py
Solve the following math problem:
What is 12 divided by 3?
Respond with only the answer.
```

Observation:

The code defines a function math_prompt_with_context that generates a math help prompt including detailed context: grade level, topic, and difficulty. It formats a message as if from a student, asking for help and an explanation of the steps. The example usage demonstrates how to create a prompt for a 5th-grade geometry problem at medium difficulty. This approach helps an AI assistant tailor its response to the student's background and needs.

Q2. One-shot vs Few-shot [5M]

• Task 1: Write:

o A one-shot prompt (give 1 example of classification).

o A few-shot prompt (give 3–4 examples).

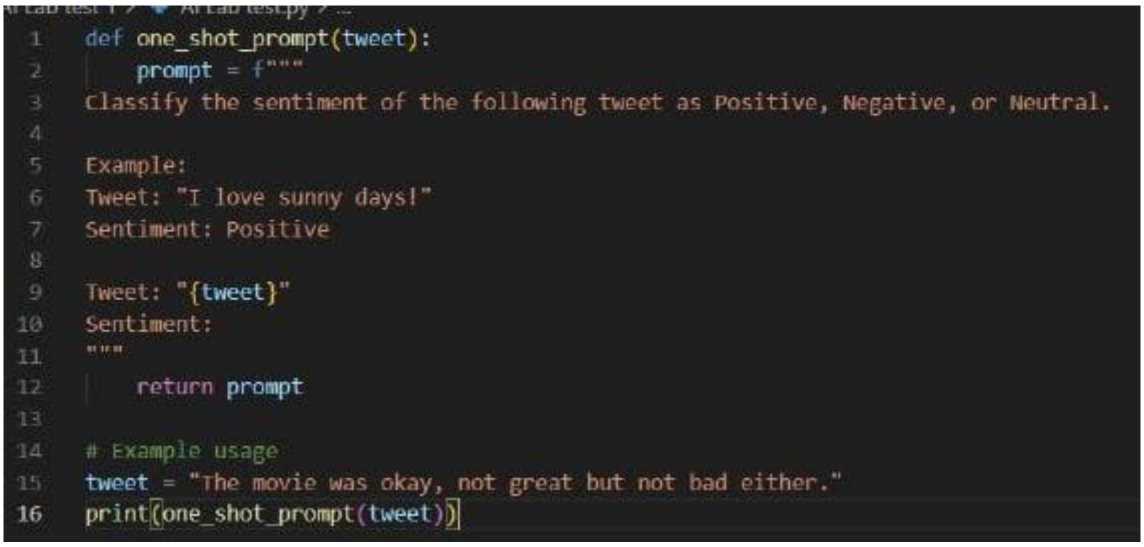
• Task 2: Compare outputs on the same set of tweets and explain the difference.

Prompt :

# Task 1: Write:

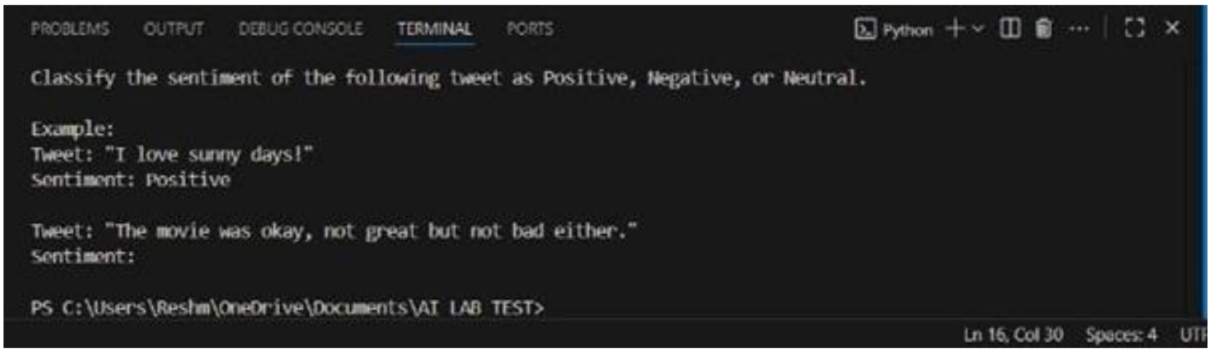
o A one-shot (give 1 example of classification).

Code:



```
def one_shot_prompt(tweet):
    prompt = f"""
Classify the sentiment of the following tweet as Positive, Negative, or Neutral.

Example:
Tweet: "I love sunny days!"
Sentiment: Positive

Tweet: "{tweet}"
Sentiment:
"""
    return prompt

# Example usage
tweet = "The movie was okay, not great but not bad either."
print(one_shot_prompt(tweet))
```

Output:



```
Classify the sentiment of the following tweet as Positive, Negative, or Neutral.

Example:
Tweet: "I love sunny days!"
Sentiment: Positive

Tweet: "The movie was okay, not great but not bad either."
Sentiment:

PS C:\Users\Reshm\OneDrive\Documents\AI LAB TEST>
```

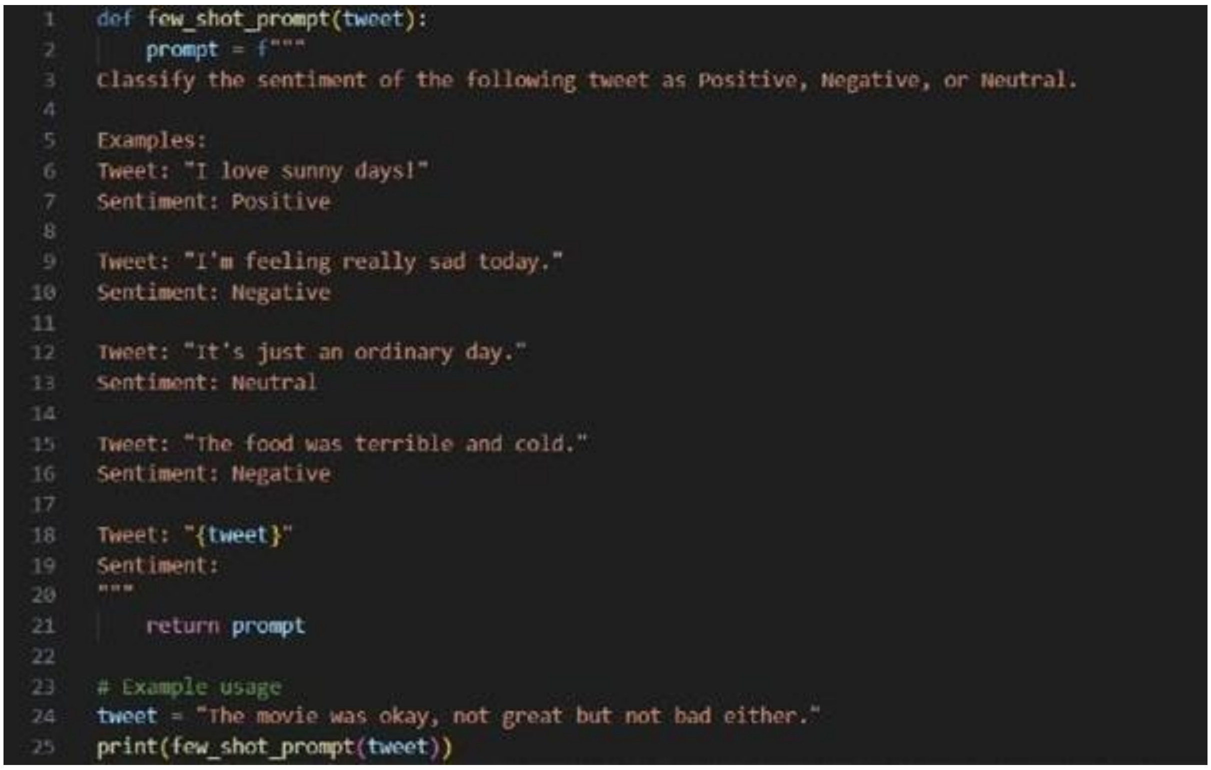
Observation:

The one-shot prompt provides a clear example of how to classify sentiment, which can help guide the model's response. However, the model's performance may still vary based on the complexity of the tweet and the clarity of the example provided.def few_shot_prompt(tweet):

Prompt :

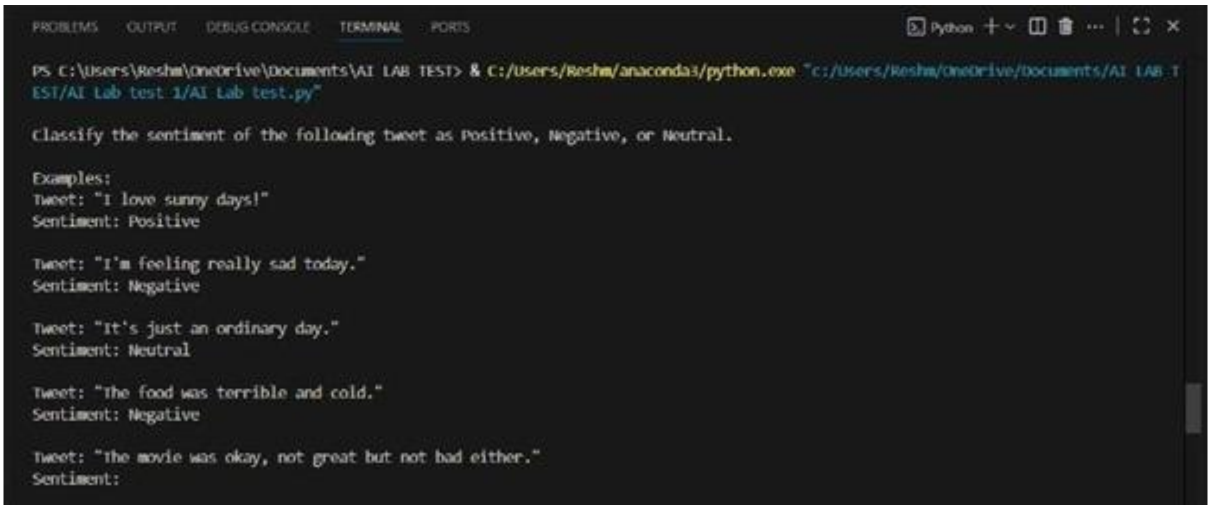
o A few-shot prompt (give 3–4 examples).

Code:



```python
def few_shot_prompt(tweet):
    prompt = f"""
Classify the sentiment of the following tweet as Positive, Negative, or Neutral.

Examples:
Tweet: "I love sunny days!"
Sentiment: Positive

Tweet: "I'm feeling really sad today."
Sentiment: Negative

Tweet: "It's just an ordinary day."
Sentiment: Neutral

Tweet: "The food was terrible and cold."
Sentiment: Negative

Tweet: "{tweet}"
Sentiment:
"""
    return prompt

# Example usage
tweet = "The movie was okay, not great but not bad either."
print(few_shot_prompt(tweet))
```

Output:



```
PS C:\Users\Reshm\OneDrive\Documents\AI LAB TEST> & C:/Users/Reshm/anaconda3/python.exe "c:/Users/Reshm/OneDrive/Documents/AI LAB T
EST/AI Lab test 1/AI Lab test.py"

Classify the sentiment of the following tweet as Positive, Negative, or Neutral.

Examples:
Tweet: "I love sunny days!"
Sentiment: Positive

Tweet: "I'm feeling really sad today."
Sentiment: Negative

Tweet: "It's just an ordinary day."
Sentiment: Neutral

Tweet: "The food was terrible and cold."
Sentiment: Negative

Tweet: "The movie was okay, not great but not bad either."
Sentiment:
```

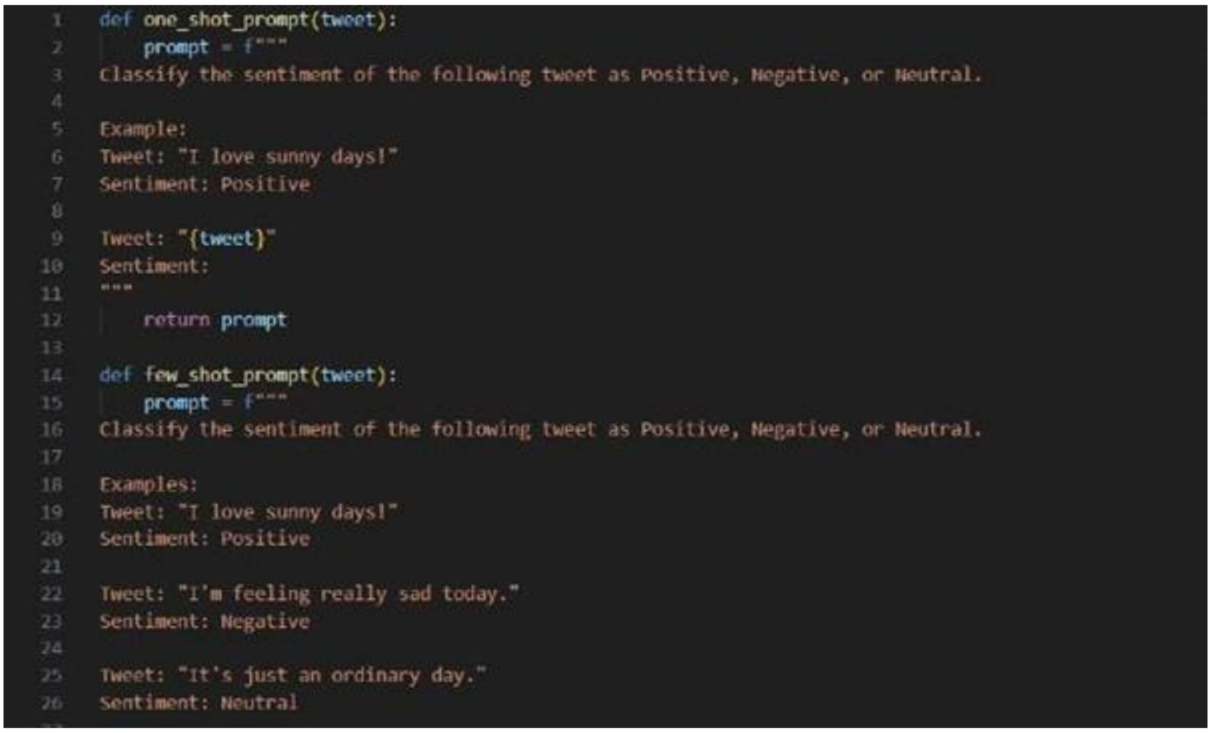
Observation:

The code defines a function few_shot_prompt(tweet) that generates a few-shot prompt for sentiment classification of tweets. The prompt includes four example tweets, each labeled with its sentiment (Positive, Negative, or Neutral). The function then appends the input tweet and leaves the sentiment blank for classification. The example usage demonstrates how to use the function by printing the prompt for the tweet "The movie was okay, not great but not bad either." This approach helps guide a language model to classify the sentiment of new tweets by providing multiple labeled examples.

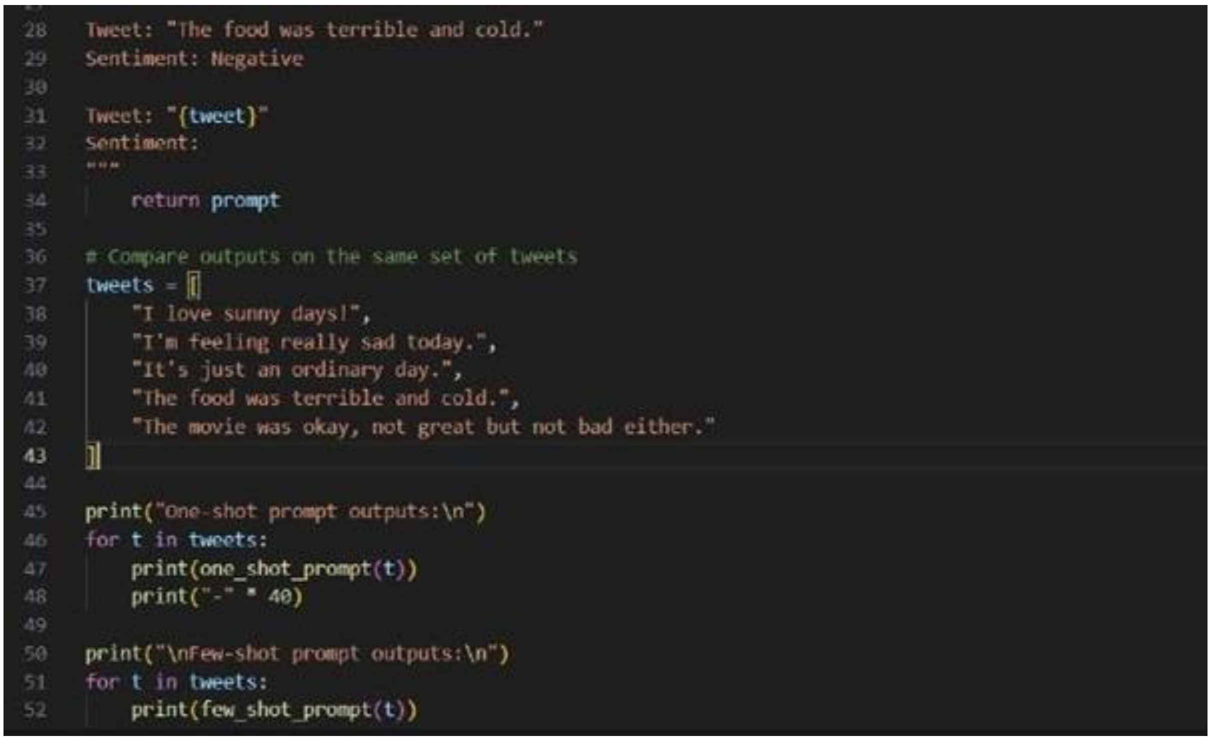
Task 2: Compare outputs on the same set of tweets and explain the difference.

Prompt :

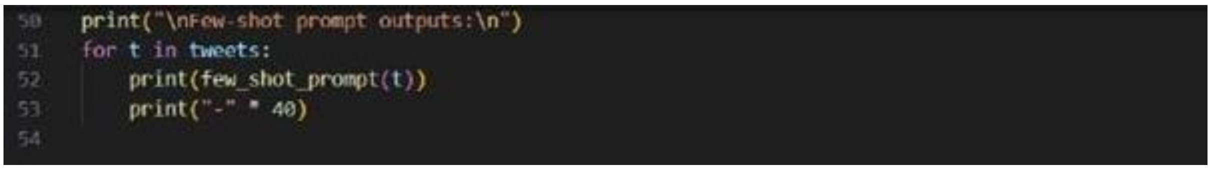
Compare outputs on the same set of tweets and explain the difference.



```python
def one_shot_prompt(tweet):
    prompt = f"""
Classify the sentiment of the following tweet as Positive, Negative, or Neutral.

Example:
Tweet: "I love sunny days!"
Sentiment: Positive

Tweet: "{tweet}"
Sentiment:
"""
    return prompt

def few_shot_prompt(tweet):
    prompt = f"""
Classify the sentiment of the following tweet as Positive, Negative, or Neutral.

Examples:
Tweet: "I love sunny days!"
Sentiment: Positive

Tweet: "I'm feeling really sad today."
Sentiment: Negative

Tweet: "It's just an ordinary day."
Sentiment: Neutral
```

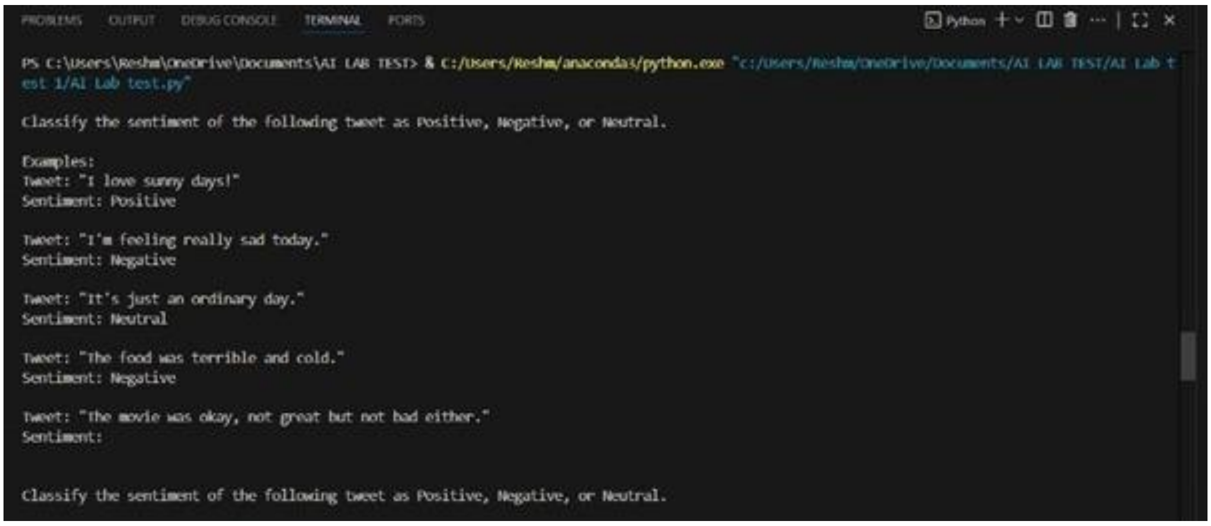


```python
Tweet: "The food was terrible and cold."
Sentiment: Negative

Tweet: "{tweet}"
Sentiment:
"""
    return prompt

# Compare outputs on the same set of tweets
tweets = [
    "I love sunny days!",
    "I'm feeling really sad today.",
    "It's just an ordinary day.",
    "The food was terrible and cold.",
    "The movie was okay, not great but not bad either."
]

print("One-shot prompt outputs:\n")
for t in tweets:
    print(one_shot_prompt(t))
    print("-" * 40)

print("\nFew-shot prompt outputs:\n")
for t in tweets:
    print(few_shot_prompt(t))
```

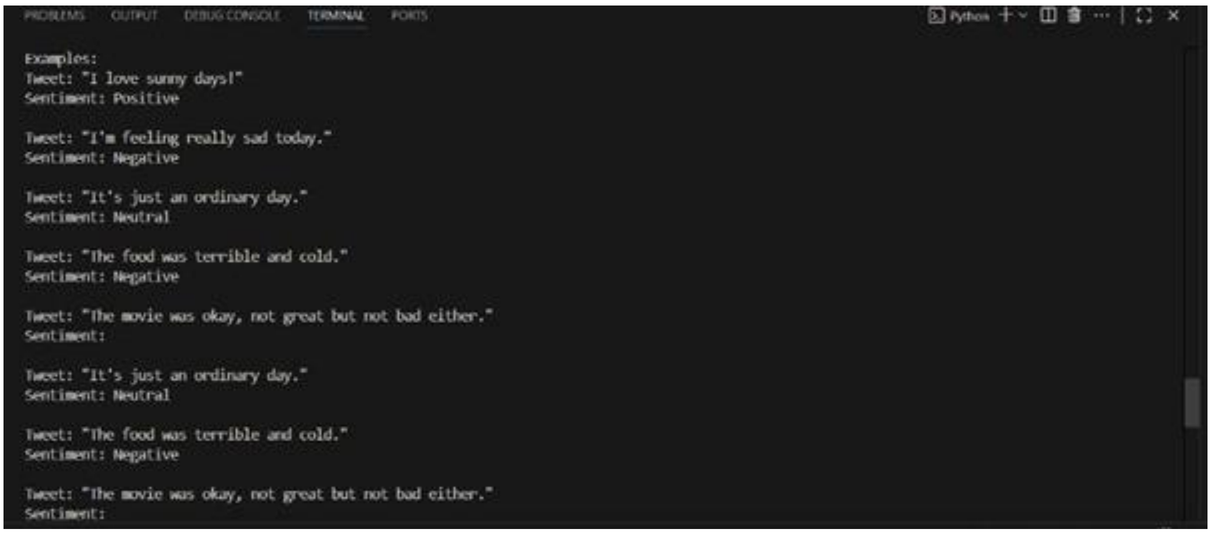


```python
print("\nFew-shot prompt outputs:\n")
for t in tweets:
    print(few_shot_prompt(t))
    print("-" * 40)
```

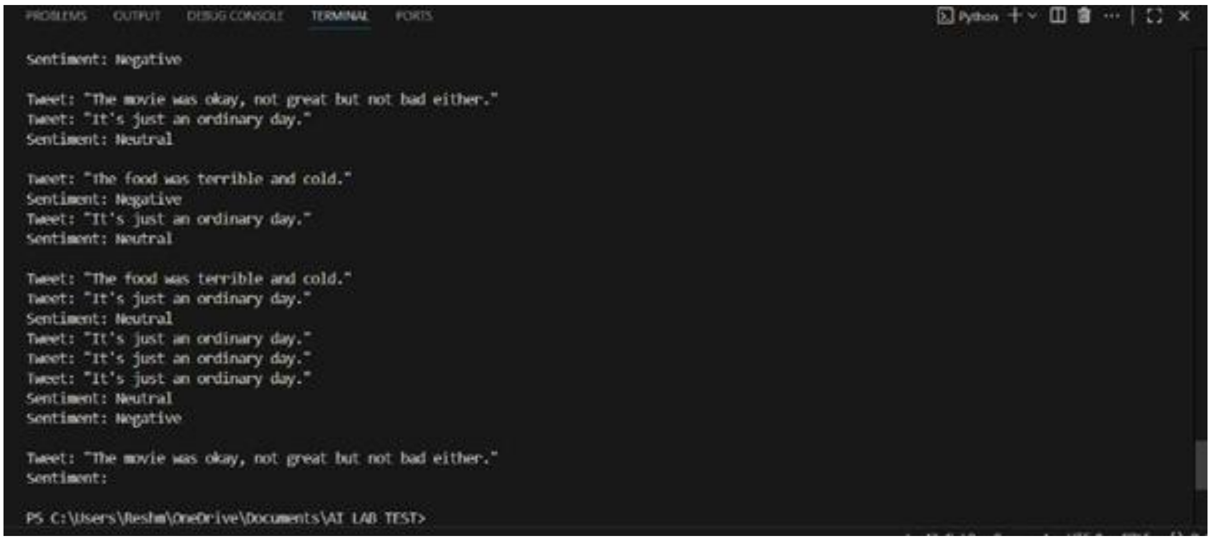
Output:



```
PS C:\Users\Reshm\OneDrive\Documents\AI LAB TEST> & C:/Users/Reshm/anaconda3/python.exe "c:/Users/Reshm/OneDrive/Documents/AI LAB TEST/AI Lab t
est 1/AI Lab test.py"

Classify the sentiment of the following tweet as Positive, Negative, or Neutral.

Examples:
Tweet: "I love sunny days!"
Sentiment: Positive

Tweet: "I'm feeling really sad today."
Sentiment: Negative

Tweet: "It's just an ordinary day."
Sentiment: Neutral

Tweet: "The food was terrible and cold."
Sentiment: Negative

Tweet: "The movie was okay, not great but not bad either."
Sentiment:


Classify the sentiment of the following tweet as Positive, Negative, or Neutral.
```



```
Examples:
Tweet: "I love sunny days!"
Sentiment: Positive

Tweet: "I'm feeling really sad today."
Sentiment: Negative

Tweet: "It's just an ordinary day."
Sentiment: Neutral

Tweet: "The food was terrible and cold."
Sentiment: Negative

Tweet: "The movie was okay, not great but not bad either."
Sentiment:

Tweet: "It's just an ordinary day."
Sentiment: Neutral

Tweet: "The food was terrible and cold."
Sentiment: Negative

Tweet: "The movie was okay, not great but not bad either."
Sentiment:
```



```
Sentiment: Negative

Tweet: "The movie was okay, not great but not bad either."
Tweet: "It's just an ordinary day."
Sentiment: Neutral

Tweet: "The food was terrible and cold."
Sentiment: Negative
Tweet: "It's just an ordinary day."
Sentiment: Neutral

Tweet: "The food was terrible and cold."
Tweet: "It's just an ordinary day."
Sentiment: Neutral
Tweet: "It's just an ordinary day."
Tweet: "It's just an ordinary day."
Tweet: "It's just an ordinary day."
Sentiment: Neutral
Sentiment: Negative

Tweet: "The movie was okay, not great but not bad either."
Sentiment:

PS C:\Users\Reshm\OneDrive\Documents\AI LAB TEST>
```

Observation :

The code defines two functions, one_shot_prompt and few_shot_prompt, to generate prompts for sentiment classification of tweets. The one_shot_prompt function provides a single example to guide the model, while the few_shot_prompt function provides four diverse examples (positive, negative, and neutral). The code then iterates over a list of tweets, printing the generated prompts for each tweet using both methods.