AI ASSISTANT CODING

ASSIGNMENT-4

Name:B.Bhargava chary

Batch:24

Ht.No:2303A51747

Q1. Zero-Shot Prompting (Basic Lab Task)

Task:

Write a Python function that classifies a given text as Spam or Not

Spam using zero-shot prompting.

Steps:

1. Construct a prompt without any examples.

2. Clearly specify the output labels.

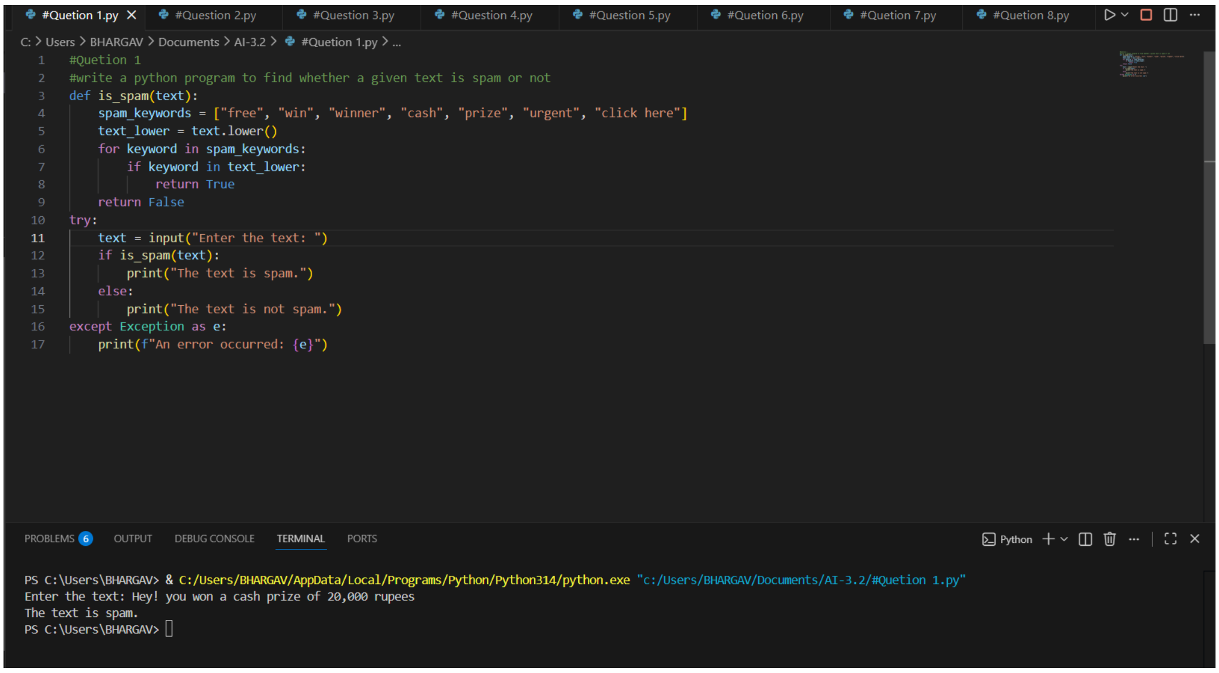
3. Display only the predicted label.

Input:

"Hey! you won a cash prize of 20,000 rupees"

Expected Output:

The text is spam.



```python
#Quetion 1
#write a python program to find whether a given text is spam or not
def is_spam(text):
    spam_keywords = ["free", "win", "winner", "cash", "prize", "urgent", "click here"]
    text_lower = text.lower()
    for keyword in spam_keywords:
        if keyword in text_lower:
            return True
    return False
try:
    text = input("Enter the text: ")
    if is_spam(text):
        print("The text is spam.")
    else:
        print("The text is not spam.")
except Exception as e:
    print(f"An error occurred: {e}")
```

```
PS C:\Users\BHARGAV> & C:/Users/BHARGAV/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/BHARGAV/Documents/AI-3.2/#Quetion 1.py"
Enter the text: Hey! you won a cash prize of 20,000 rupees
The text is spam.
PS C:\Users\BHARGAV>
```

Q2. One-Shot Prompting (Emotion detection)

Task:

Write a Python program that detects the emotion of a sentence

using one-shot prompting.

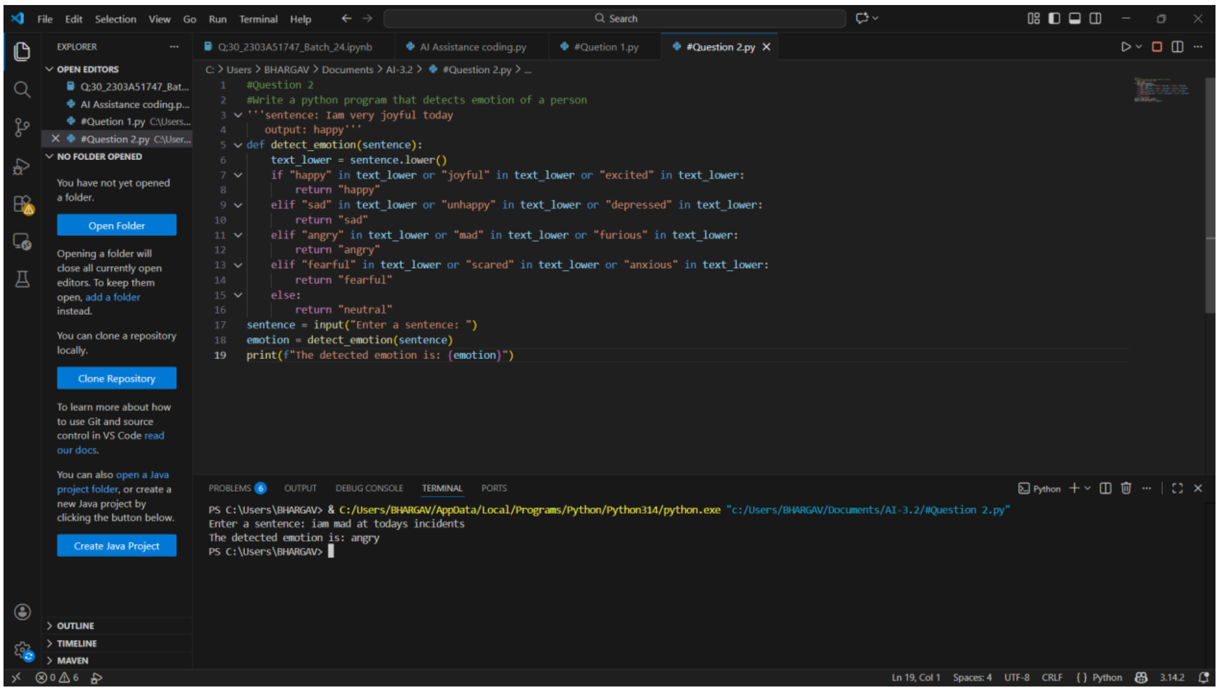
Emotions: [‘happy', 'sad', 'angry', 'excited', 'nervous', ’neutral’]

Steps:

1. Provide one labeled example inside the prompt.

2. Take a sentence as input.

3. Print the predicted emotion



```python
#Question 2
#Write a python program that detects emotion of a person
'''sentence: Iam very joyful today
  output: happy'''
def detect_emotion(sentence):
    text_lower = sentence.lower()
    if "happy" in text_lower or "joyful" in text_lower or "excited" in text_lower:
        return "happy"
    elif "sad" in text_lower or "unhappy" in text_lower or "depressed" in text_lower:
        return "sad"
    elif "angry" in text_lower or "mad" in text_lower or "furious" in text_lower:
        return "angry"
    elif "fearful" in text_lower or "scared" in text_lower or "anxious" in text_lower:
        return "fearful"
    else:
        return "neutral"
sentence = input("Enter a sentence: ")
emotion = detect_emotion(sentence)
print(f"The detected emotion is: {emotion}")
```

```
PS C:\Users\BHARGAV> & C:/Users/BHARGAV/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/BHARGAV/Documents/AI-3.2/#Question 2.py"
Enter a sentence: iam mad at todays incidents
The detected emotion is: angry
PS C:\Users\BHARGAV>
```

Q3. Few-Shot Prompting (Student Grading Based on Marks)

Task:

Write a Python program that predicts a student’s grade based on

marks using few-shot prompting.

Grades:

['A', 'B', 'C', 'D', 'F']

Grading Criteria (to be inferred from examples):

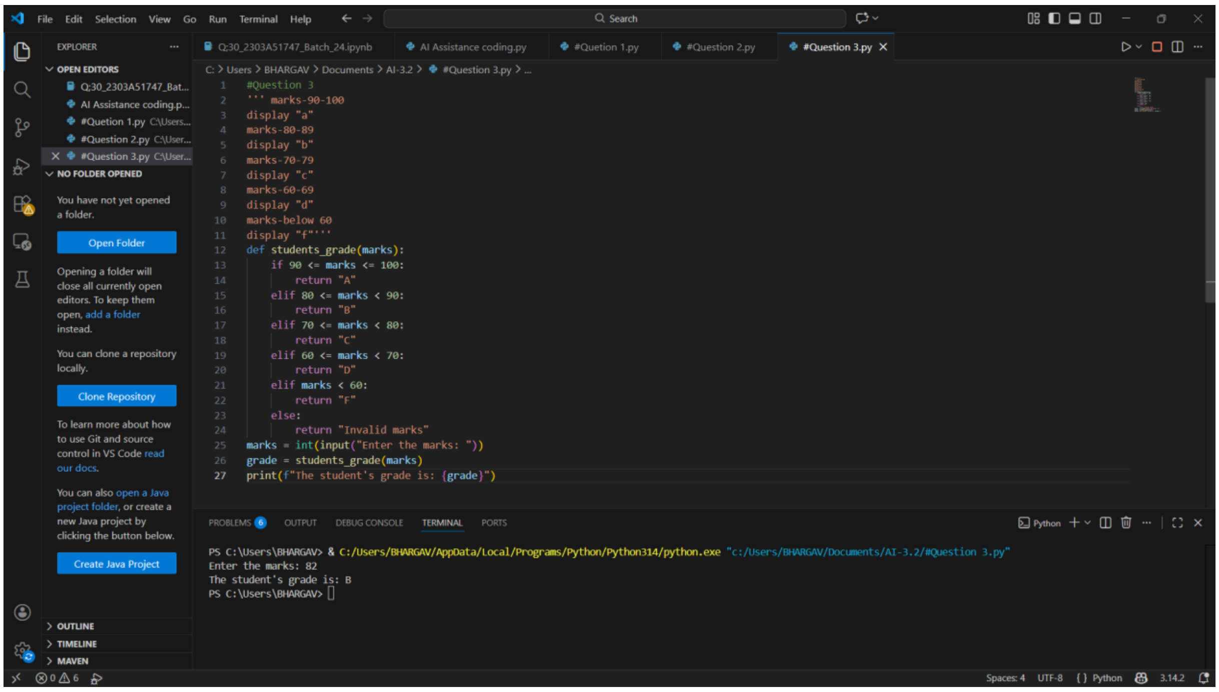
• 90–100 → A

• 80–89 → B

• 70–79 → C

• 60–69 → D

• Below 60 → F

• 

```python
#Question 3
''' marks-90-100
display "a"
marks-80-89
display "b"
marks-70-79
display "c"
marks-60-69
display "d"
marks-below 60
display "f"'''
def students_grade(marks):
    if 90 <= marks <= 100:
        return "A"
    elif 80 <= marks < 90:
        return "B"
    elif 70 <= marks < 80:
        return "C"
    elif 60 <= marks < 70:
        return "D"
    elif marks < 60:
        return "F"
    else:
        return "Invalid marks"
marks = int(input("Enter the marks: "))
grade = students_grade(marks)
print(f"The student's grade is: {grade}")
```

```
PS C:\Users\BHARGAV> & C:/Users/BHARGAV/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/BHARGAV/Documents/AI-3.2/#Question 3.py"
Enter the marks: 82
The student's grade is: B
PS C:\Users\BHARGAV>
```

Q4. Multi-Shot Prompting (Indian Zodiac Sign Prediction using

Month Name)

Task:

Write a Python program that predicts a person's Indian Zodiac sign

(Rashi) based on the month of birth (month name) using multi-shot

prompting.

Indian Zodiac Order (Simplified Month-Based Model): The Indian

Zodiac cycle starts in March with Mesha and follows this order:

March → Mesha

April → Vrishabha

May → Mithuna

June → Karka

July → Simha

August → Kanya

September → Tula

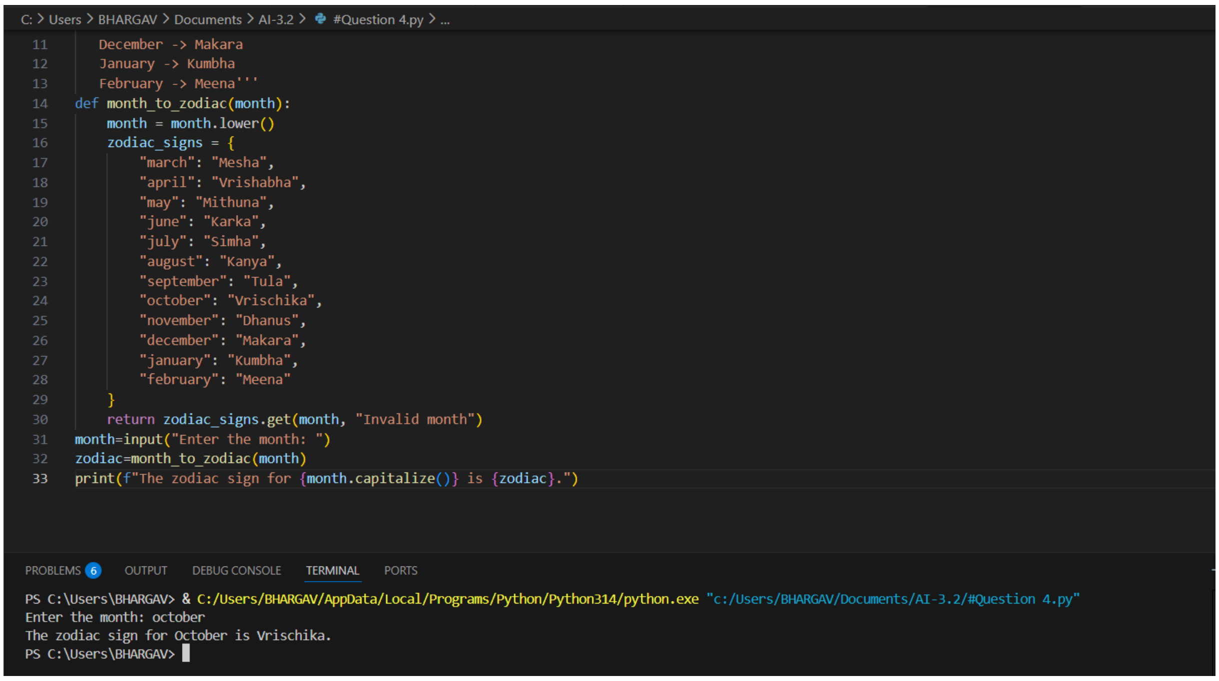
October → Vrischika

November → Dhanu

December → Makara

January → Kumbha

February → Meena



C: > Users > BHARGAV > Documents > AI-3.2 > #Question 4.py > ...

```python
    December -> Makara
    January -> Kumbha
    February -> Meena'''
def month_to_zodiac(month):
    month = month.lower()
    zodiac_signs = {
        "march": "Mesha",
        "april": "Vrishabha",
        "may": "Mithuna",
        "june": "Karka",
        "july": "Simha",
        "august": "Kanya",
        "september": "Tula",
        "october": "Vrischika",
        "november": "Dhanus",
        "december": "Makara",
        "january": "Kumbha",
        "february": "Meena"
    }
    return zodiac_signs.get(month, "Invalid month")
month=input("Enter the month: ")
zodiac=month_to_zodiac(month)
print(f"The zodiac sign for {month.capitalize()} is {zodiac}.")
```

```
PS C:\Users\BHARGAV> & C:/Users/BHARGAV/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/BHARGAV/Documents/AI-3.2/#Question 4.py"
Enter the month: october
The zodiac sign for October is Vrischika.
PS C:\Users\BHARGAV> 
```

Q5. Result Analysis Based on Marks

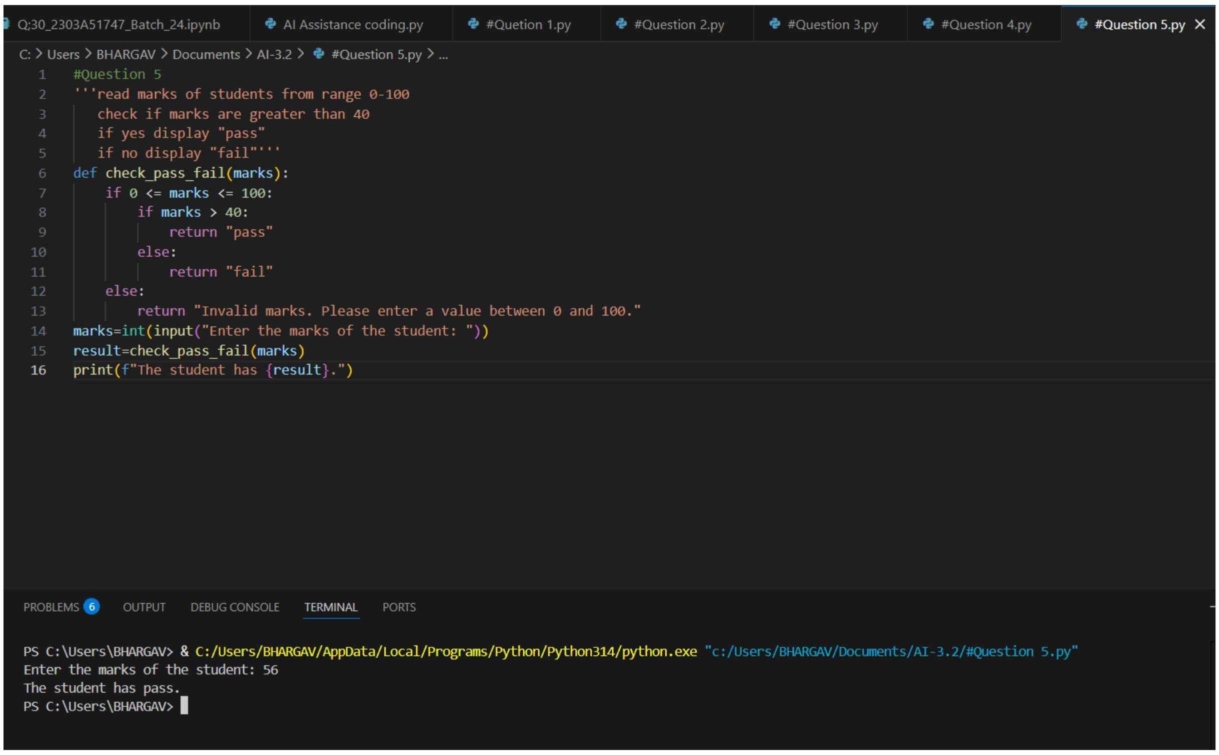
Task: Write a Python program that determines whether a student

Passes or Fails based on marks using Chain-of-Thought (CoT)

prompting.

Result Categories:

['Pass', 'Fail']



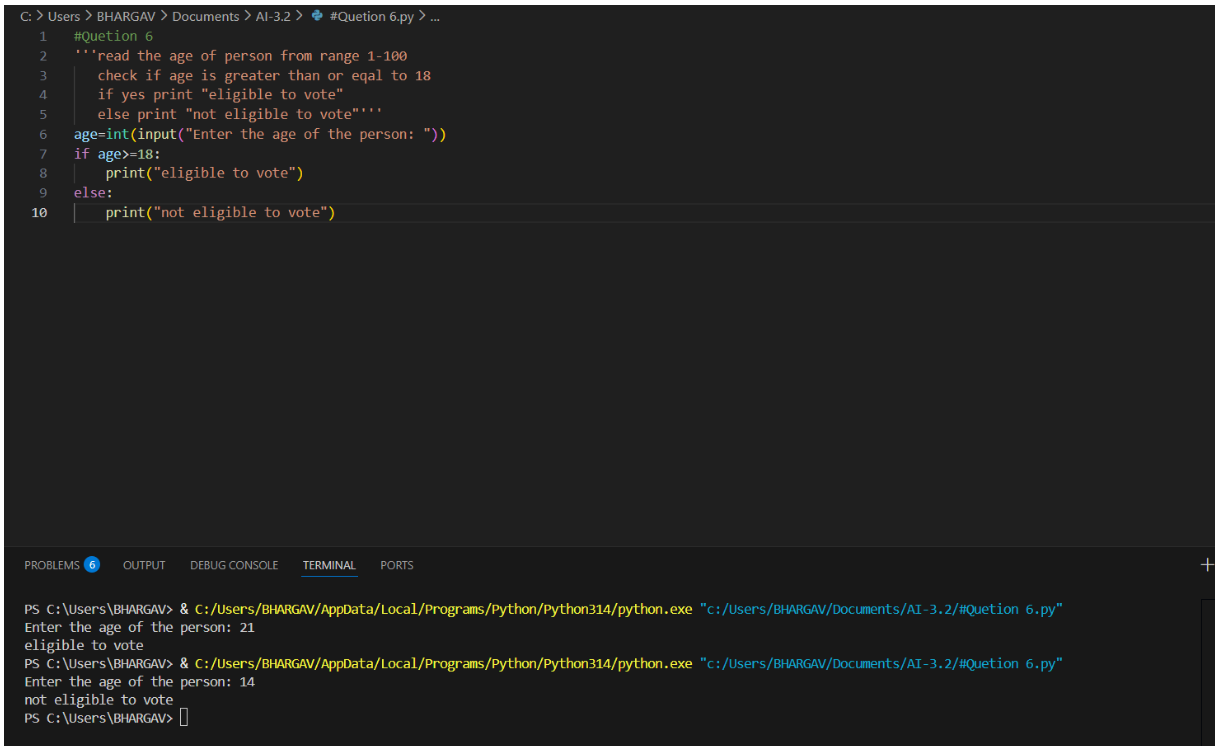
```python
#Question 5
'''read marks of students from range 0-100
    check if marks are greater than 40
    if yes display "pass"
    if no display "fail"'''
def check_pass_fail(marks):
    if 0 <= marks <= 100:
        if marks > 40:
            return "pass"
        else:
            return "fail"
    else:
        return "Invalid marks. Please enter a value between 0 and 100."
marks=int(input("Enter the marks of the student: "))
result=check_pass_fail(marks)
print(f"The student has {result}.")
```

```
PS C:\Users\BHARGAV> & C:/Users/BHARGAV/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/BHARGAV/Documents/AI-3.2/#Question 5.py"
Enter the marks of the student: 56
The student has pass.
PS C:\Users\BHARGAV> 
```

Q6 Voting Eligibility Check (Chain-of-Thought Prompting) Task:

Write a Python program that determines whether a person is

eligible to vote using Chain-of-Thought (CoT) prompting.


C: > Users > BHARGAV > Documents > AI-3.2 > #Quetion 6.py > ...
PROBLEMS 6
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```python
#Quetion 6
'''read the age of person from range 1-100
   check if age is greater than or eqal to 18
   if yes print "eligible to vote"
   else print "not eligible to vote"'''
age=int(input("Enter the age of the person: "))
if age>=18:
    print("eligible to vote")
else:
    print("not eligible to vote")
```

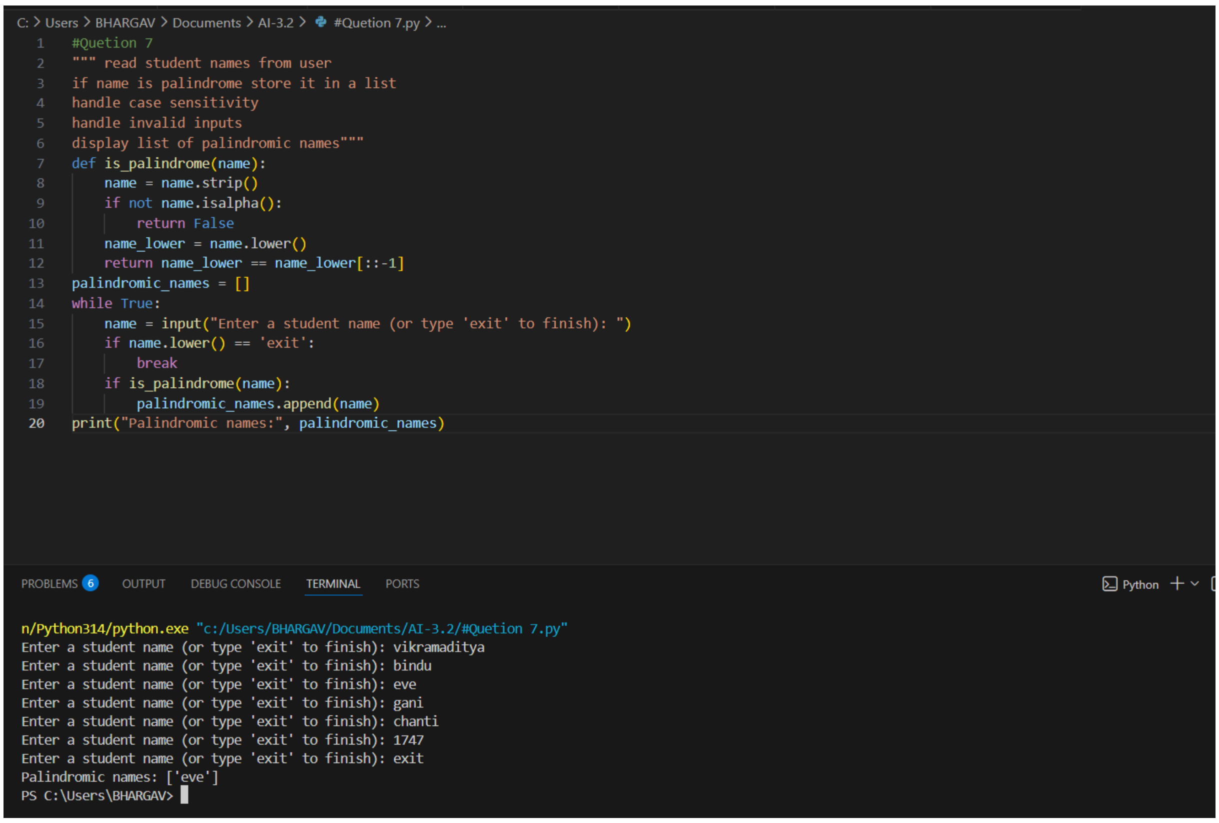
```
PS C:\Users\BHARGAV> & C:/Users/BHARGAV/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/BHARGAV/Documents/AI-3.2/#Quetion 6.py"
Enter the age of the person: 21
eligible to vote
PS C:\Users\BHARGAV> & C:/Users/BHARGAV/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/BHARGAV/Documents/AI-3.2/#Quetion 6.py"
Enter the age of the person: 14
not eligible to vote
PS C:\Users\BHARGAV> 
```

Q7 Prompt Chaining (String Processing – Palindrome Names) Task:

Write a Python program that uses the prompt chaining technique

to identify palindrome names from a list of student

names.



C: > Users > BHARGAV > Documents > AI-3.2 > #Quetion 7.py > ...

```python
#Quetion 7
""" read student names from user
if name is palindrome store it in a list
handle case sensitivity
handle invalid inputs
display list of palindromic names"""
def is_palindrome(name):
    name = name.strip()
    if not name.isalpha():
        return False
    name_lower = name.lower()
    return name_lower == name_lower[::-1]
palindromic_names = []
while True:
    name = input("Enter a student name (or type 'exit' to finish): ")
    if name.lower() == 'exit':
        break
    if is_palindrome(name):
        palindromic_names.append(name)
print("Palindromic names:", palindromic_names)
```

```
n/Python314/python.exe "c:/Users/BHARGAV/Documents/AI-3.2/#Quetion 7.py"
Enter a student name (or type 'exit' to finish): vikramaditya
Enter a student name (or type 'exit' to finish): bindu
Enter a student name (or type 'exit' to finish): eve
Enter a student name (or type 'exit' to finish): gani
Enter a student name (or type 'exit' to finish): chanti
Enter a student name (or type 'exit' to finish): 1747
Enter a student name (or type 'exit' to finish): exit
Palindromic names: ['eve']
PS C:\Users\BHARGAV>
```

Q8 Prompt Chaining (String Processing – Word Length

Analysis)

Task: Write a Python program that uses prompt chaining to analyze

a list of words. In the first prompt, generate a list of words. In the

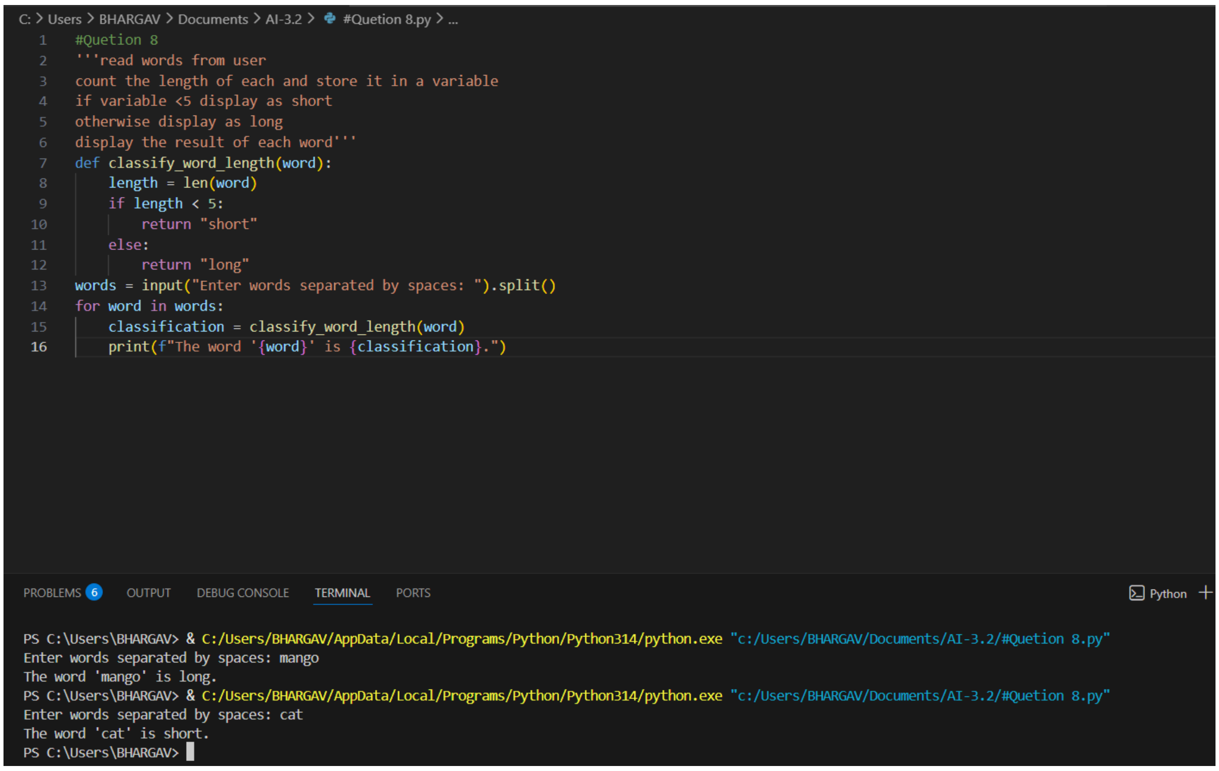
second prompt, traverse the list and calculate the length of each

word. In the third prompt, use the output of the previous step to

determine whether each word is Short (length less than 5) or Long

(length greater than or equal to 5), and display the result for

each word



C: > Users > BHARGAV > Documents > AI-3.2 > #Quetion 8.py > ...

```python
#Quetion 8
'''read words from user
count the length of each and store it in a variable
if variable <5 display as short
otherwise display as long
display the result of each word'''
def classify_word_length(word):
    length = len(word)
    if length < 5:
        return "short"
    else:
        return "long"
words = input("Enter words separated by spaces: ").split()
for word in words:
    classification = classify_word_length(word)
    print(f"The word '{word}' is {classification}.")
```

```
PS C:\Users\BHARGAV> & C:/Users/BHARGAV/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/BHARGAV/Documents/AI-3.2/#Quetion 8.py"
Enter words separated by spaces: mango
The word 'mango' is long.
PS C:\Users\BHARGAV> & C:/Users/BHARGAV/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/BHARGAV/Documents/AI-3.2/#Quetion 8.py"
Enter words separated by spaces: cat
The word 'cat' is short.
PS C:\Users\BHARGAV>
```