# AI Assisted Coding

# Assignment-4

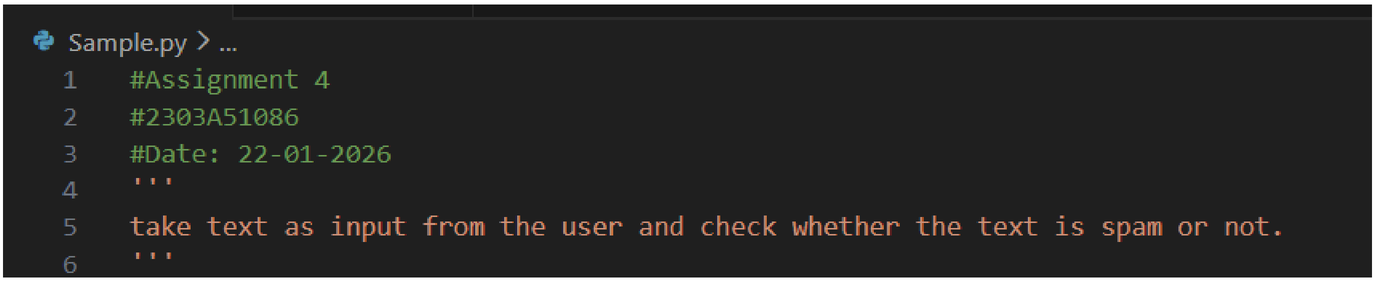
**Name : Abhiram Yadav**

**Hallticket:2303A51086**

**Batch-02**

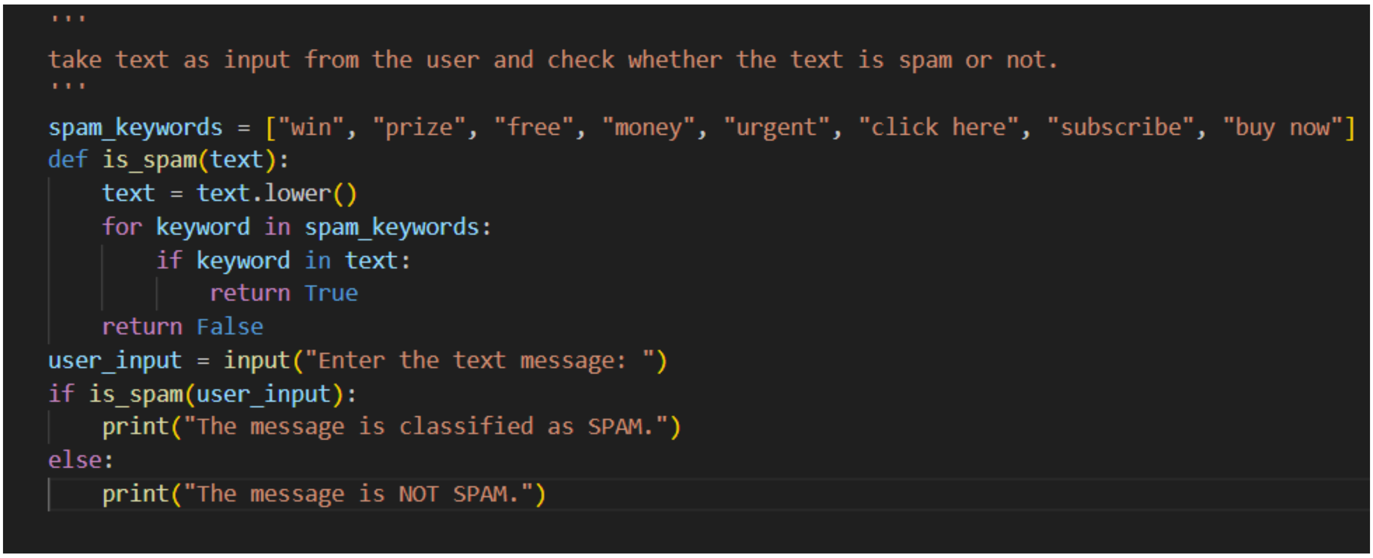
**Question:01** Zero-Shot Prompting (Basic Lab Task)

- **Prompt given for copilot**



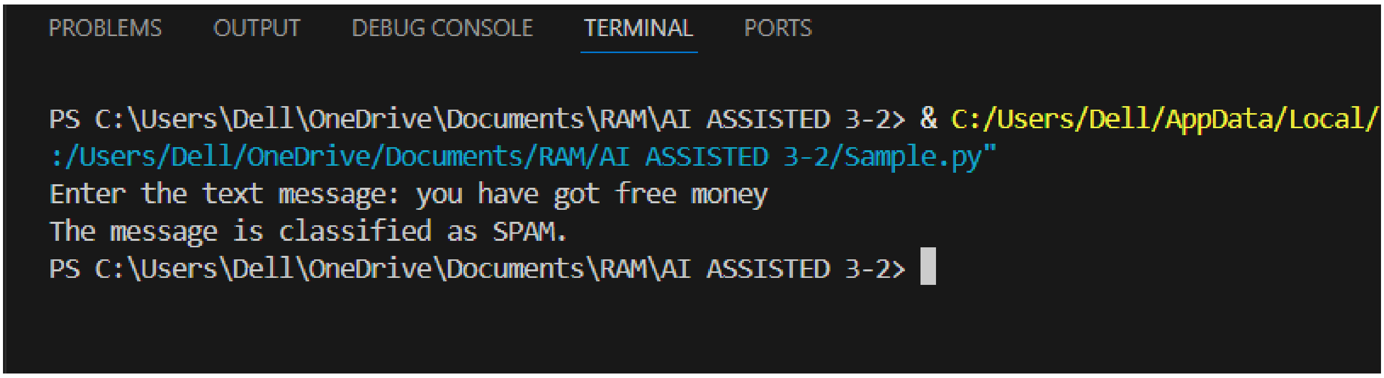
```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
take text as input from the user and check whether the text is spam or not.
'''
```

- **A working Python program generated with Copilot assistance**

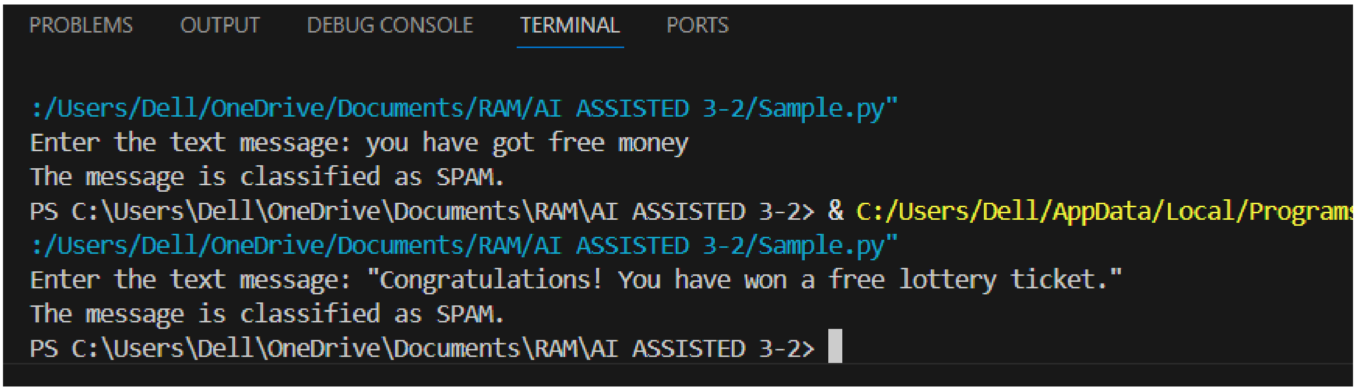


```python
'''
take text as input from the user and check whether the text is spam or not.
'''
spam_keywords = ["win", "prize", "free", "money", "urgent", "click here", "subscribe", "buy now"]
def is_spam(text):
    text = text.lower()
    for keyword in spam_keywords:
        if keyword in text:
            return True
    return False
user_input = input("Enter the text message: ")
if is_spam(user_input):
    print("The message is classified as SPAM.")
else:
    print("The message is NOT SPAM.")
```

- **Sample input/output screenshots**



```
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> & C:/Users/Dell/AppData/Local/
:/Users/Dell/OneDrive/Documents/RAM/AI ASSISTED 3-2/Sample.py"
Enter the text message: you have got free money
The message is classified as SPAM.
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2>
```



```
:/Users/Dell/OneDrive/Documents/RAM/AI ASSISTED 3-2/Sample.py"
Enter the text message: you have got free money
The message is classified as SPAM.
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> & C:/Users/Dell/AppData/Local/Programs
:/Users/Dell/OneDrive/Documents/RAM/AI ASSISTED 3-2/Sample.py"
Enter the text message: "Congratulations! You have won a free lottery ticket."
The message is classified as SPAM.
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2>
```

**Prompt:**

'''

take text as input from the user and check whether the text is spam or not.

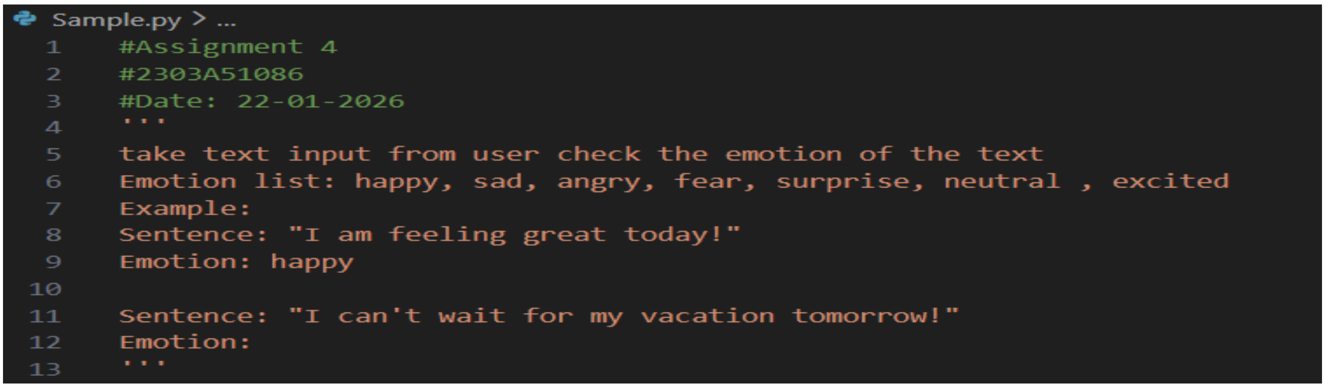
'''

**Code generated:**

```
spam_keywords = ["win", "prize", "free", "money", "urgent", "click here", "subscribe", "buy now"]
def is_spam(text):
    text = text.lower()
    for keyword in spam_keywords:
        if keyword in text:
            return True
    return False
user_input = input("Enter the text message: ")
if is_spam(user_input):
    print("The message is classified as SPAM.")
else:
    print("The message is NOT SPAM.")
```

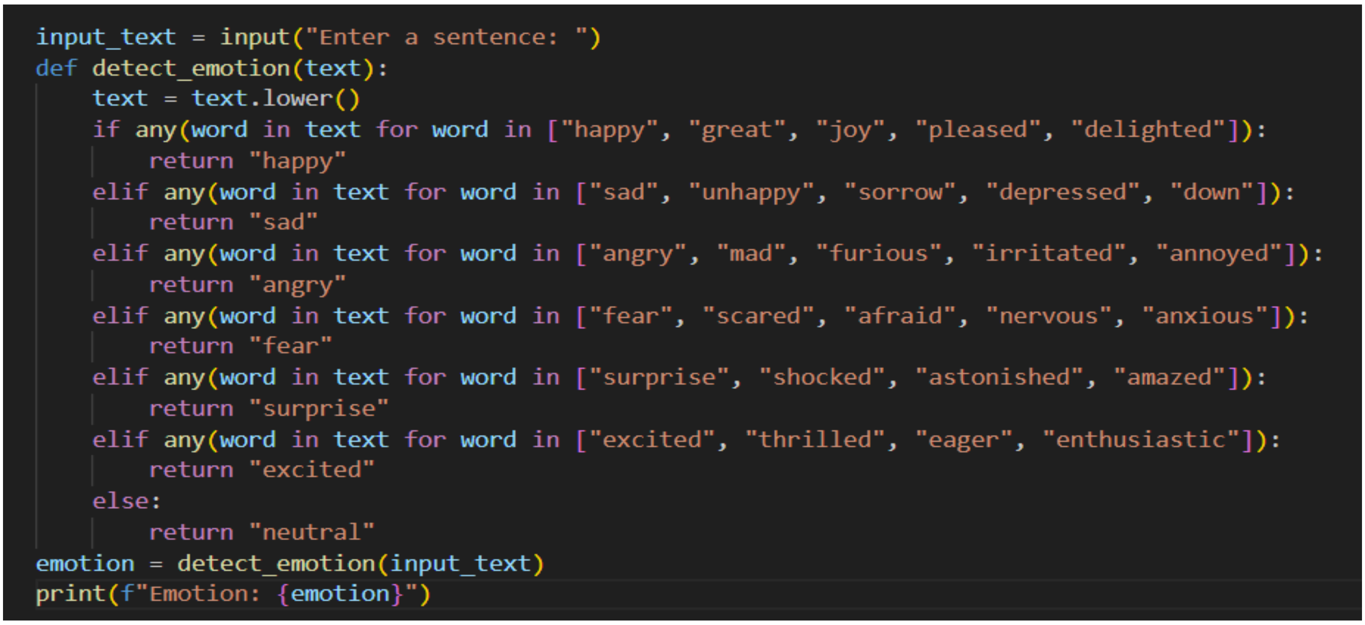
## **Question-02** One-Shot Prompting (Emotion detection)

**Prompt:**



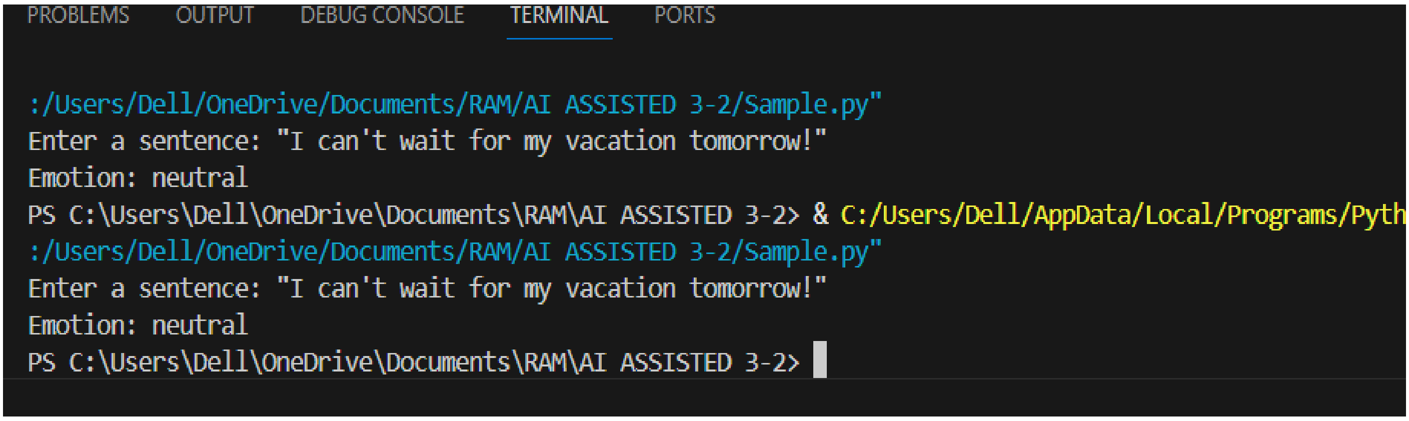
```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
take text input from user check the emotion of the text
Emotion list: happy, sad, angry, fear, surprise, neutral , excited
Example:
Sentence: "I am feeling great today!"
Emotion: happy

Sentence: "I can't wait for my vacation tomorrow!"
Emotion:
'''
```

- **A working Python program generated with Copilot assistance**



```python
input_text = input("Enter a sentence: ")
def detect_emotion(text):
    text = text.lower()
    if any(word in text for word in ["happy", "great", "joy", "pleased", "delighted"]):
        return "happy"
    elif any(word in text for word in ["sad", "unhappy", "sorrow", "depressed", "down"]):
        return "sad"
    elif any(word in text for word in ["angry", "mad", "furious", "irritated", "annoyed"]):
        return "angry"
    elif any(word in text for word in ["fear", "scared", "afraid", "nervous", "anxious"]):
        return "fear"
    elif any(word in text for word in ["surprise", "shocked", "astonished", "amazed"]):
        return "surprise"
    elif any(word in text for word in ["excited", "thrilled", "eager", "enthusiastic"]):
        return "excited"
    else:
        return "neutral"
emotion = detect_emotion(input_text)
print(f"Emotion: {emotion}")
```

- **Screenshot(s) showing: Input/Output**



```
:/Users/Dell/OneDrive/Documents/RAM/AI ASSISTED 3-2/Sample.py"
Enter a sentence: "I can't wait for my vacation tomorrow!"
Emotion: neutral
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> & C:/Users/Dell/AppData/Local/Programs/Pyth
:/Users/Dell/OneDrive/Documents/RAM/AI ASSISTED 3-2/Sample.py"
Enter a sentence: "I can't wait for my vacation tomorrow!"
Emotion: neutral
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> 
```

**Prompt:**

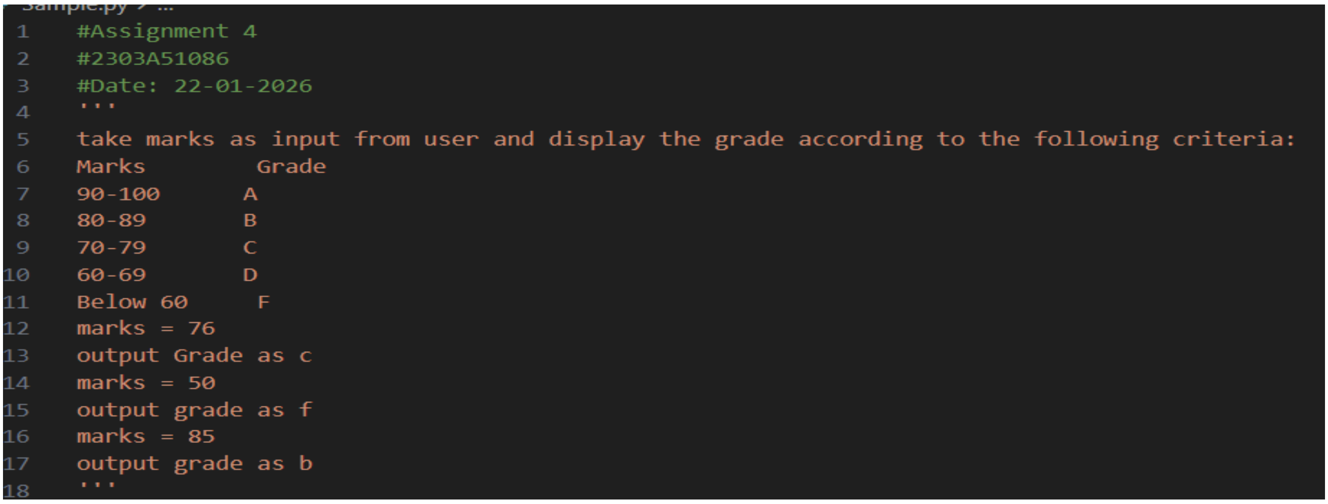
```
'''
take text input from user check the emotion of the text
Emotion list: happy, sad, angry, fear, surprise, neutral , excited
Example:
Sentence: "I am feeling great today!"
Emotion: happy

Sentence: "I can't wait for my vacation tomorrow!"
Emotion:
'''
```

**Code Generated:**

```
input_text = input("Enter a sentence: ")
def detect_emotion(text):
    text = text.lower()
    if any(word in text for word in ["happy", "great", "joy", "pleased", "delighted"]):
        return "happy"
    elif any(word in text for word in ["sad", "unhappy", "sorrow", "depressed", "down"]):
        return "sad"
    elif any(word in text for word in ["angry", "mad", "furious", "irritated", "annoyed"]):
        return "angry"
    elif any(word in text for word in ["fear", "scared", "afraid", "nervous", "anxious"]):
        return "fear"
    elif any(word in text for word in ["surprise", "shocked", "astonished", "amazed"]):
        return "surprise"
    elif any(word in text for word in ["excited", "thrilled", "eager", "enthusiastic"]):
        return "excited"
    else:
        return "neutral"
emotion = detect_emotion(input_text)
print(f"Emotion: {emotion}")
```

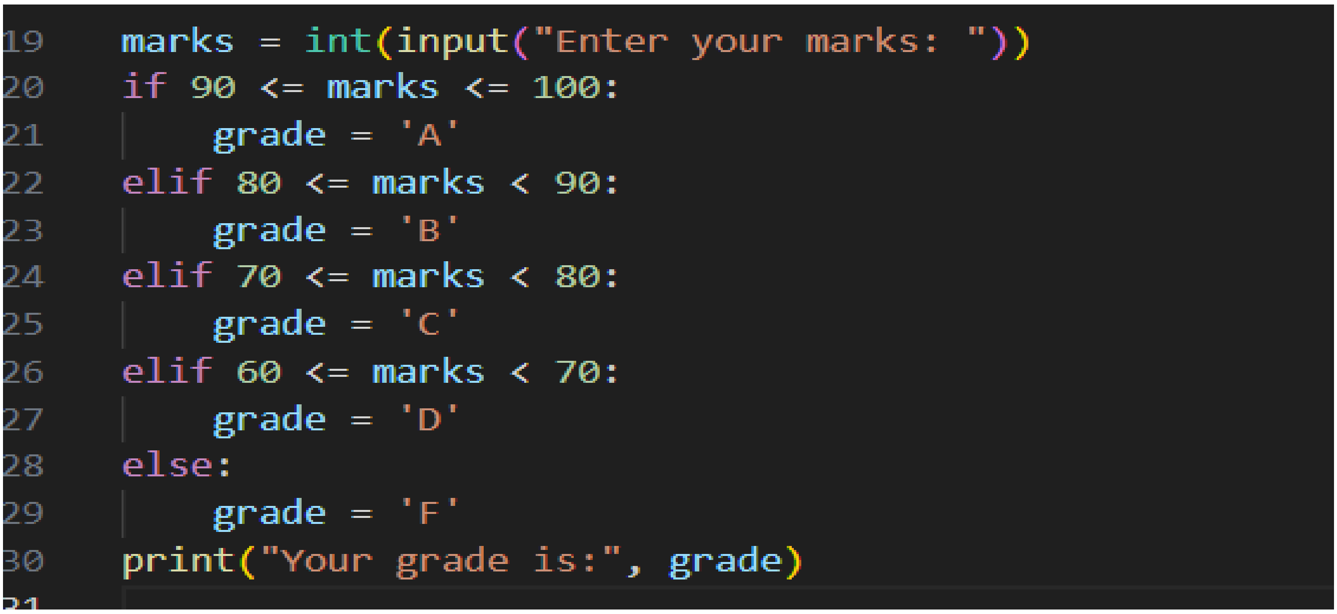
**Question-03** Few-Shot Prompting (Student Grading Based on Marks)

**Prompt:**



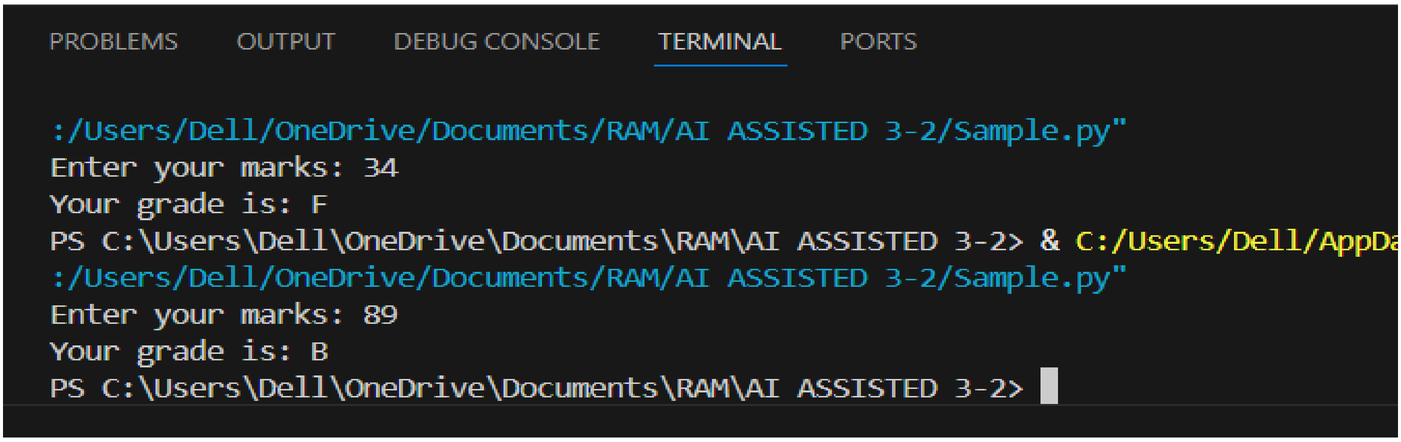
```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
take marks as input from user and display the grade according to the following criteria:
Marks        Grade
90-100      A
80-89       B
70-79       C
60-69       D
Below 60     F
marks = 76
output Grade as c
marks = 50
output grade as f
marks = 85
output grade as b
'''
```

**Code given by copilot :**



```python
marks = int(input("Enter your marks: "))
if 90 <= marks <= 100:
    grade = 'A'
elif 80 <= marks < 90:
    grade = 'B'
elif 70 <= marks < 80:
    grade = 'C'
elif 60 <= marks < 70:
    grade = 'D'
else:
    grade = 'F'
print("Your grade is:", grade)
```

- **Sample input/output screenshots**



```
:/Users/Dell/OneDrive/Documents/RAM/AI ASSISTED 3-2/Sample.py"
Enter your marks: 34
Your grade is: F
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> & C:/Users/Dell/AppDa
:/Users/Dell/OneDrive/Documents/RAM/AI ASSISTED 3-2/Sample.py"
Enter your marks: 89
Your grade is: B
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2>
```

**Prompt:**

```
'''
take marks as input from user and display the grade according to the following criteria:
Marks       Grade
90-100      A
80-89       B
70-79       C
60-69       D
Below 60    F
marks = 76
output Grade as c
marks = 50
output grade as f
marks = 85
output grade as b
'''
```

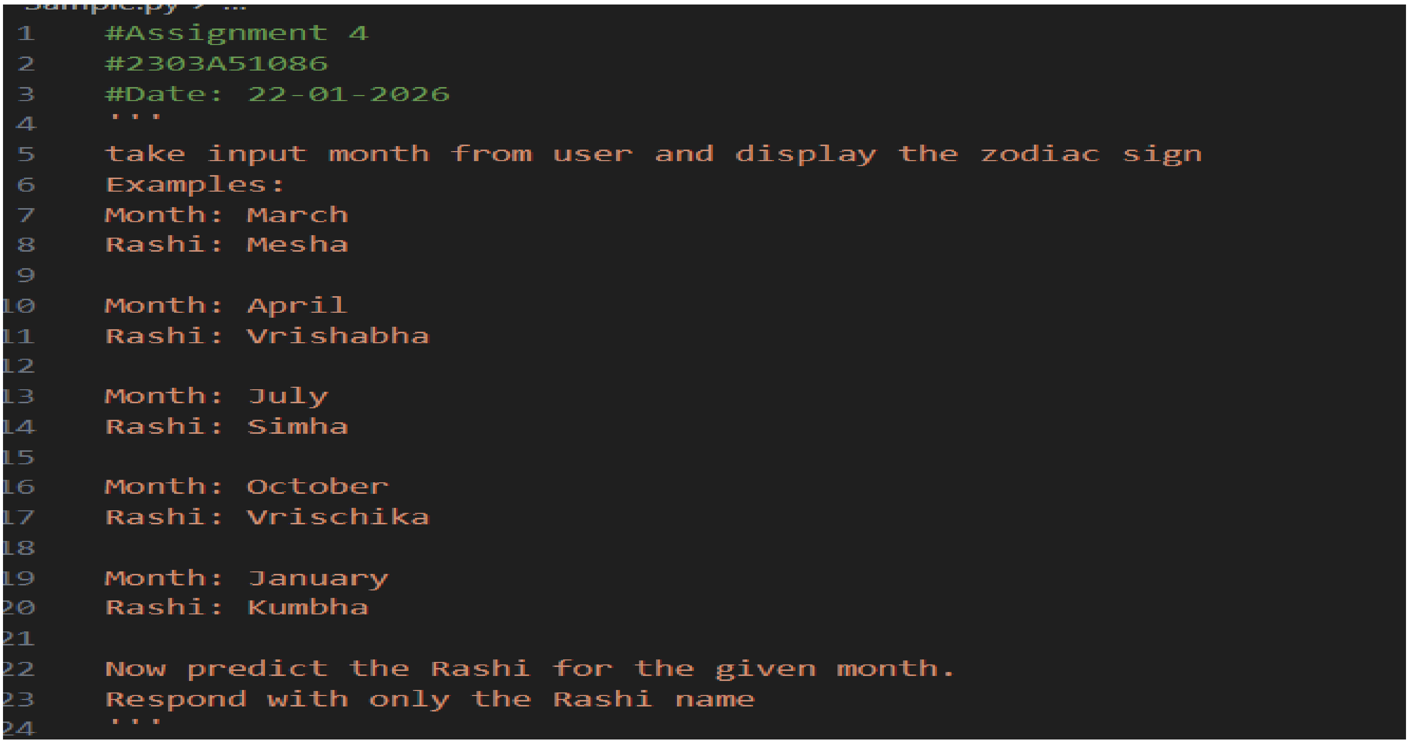
**Code Generated:**

```
marks = int(input("Enter your marks: "))
if 90 <= marks <= 100:
    grade = 'A'
elif 80 <= marks < 90:
    grade = 'B'
elif 70 <= marks < 80:
    grade = 'C'
elif 60 <= marks < 70:
    grade = 'D'
else:
    grade = 'F'
print("Your grade is:", grade)
```

**Question 4:**

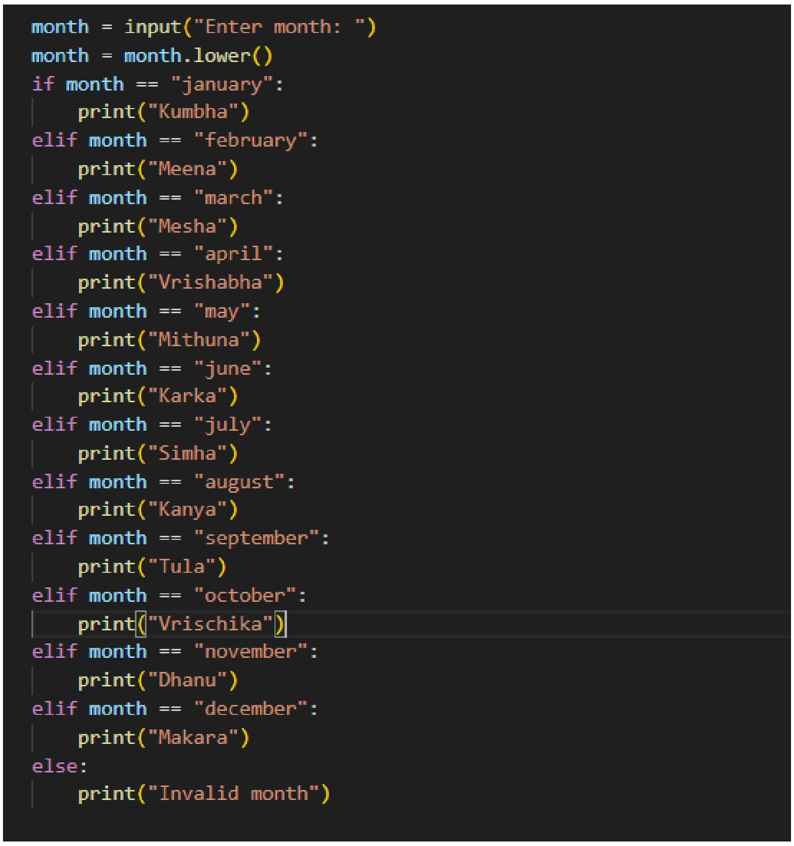
Multi-Shot Prompting (Indian Zodiac Sign Prediction using Month Name)

**Prompt:**



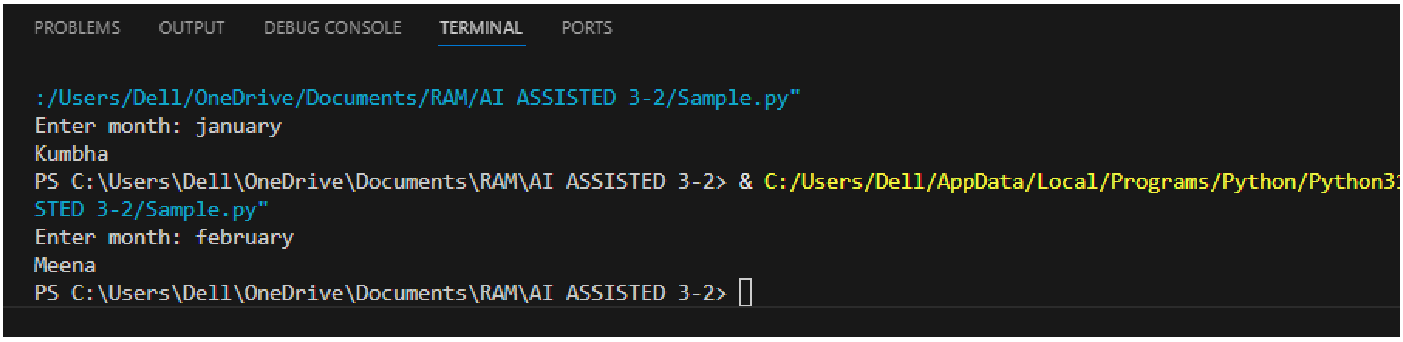
```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
take input month from user and display the zodiac sign
Examples:
Month: March
Rashi: Mesha

Month: April
Rashi: Vrishabha

Month: July
Rashi: Simha

Month: October
Rashi: Vrischika

Month: January
Rashi: Kumbha

Now predict the Rashi for the given month.
Respond with only the Rashi name
'''
```

**Code given by copilot :**



```python
month = input("Enter month: ")
month = month.lower()
if month == "january":
    print("Kumbha")
elif month == "february":
    print("Meena")
elif month == "march":
    print("Mesha")
elif month == "april":
    print("Vrishabha")
elif month == "may":
    print("Mithuna")
elif month == "june":
    print("Karka")
elif month == "july":
    print("Simha")
elif month == "august":
    print("Kanya")
elif month == "september":
    print("Tula")
elif month == "october":
    print("Vrischika")
elif month == "november":
    print("Dhanu")
elif month == "december":
    print("Makara")
else:
    print("Invalid month")
```

- **Sample input/output screenshots**



```
:/Users/Dell/OneDrive/Documents/RAM/AI ASSISTED 3-2/Sample.py"
Enter month: january
Kumbha
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> & C:/Users/Dell/AppData/Local/Programs/Python/Python3
STED 3-2/Sample.py"
Enter month: february
Meena
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> 
```

### Prompt:

'''

take input month from user and display the zodiac sign
Examples:
Month: March
Rashi: Mesha

Month: April
Rashi: Vrishabha

Month: July
Rashi: Simha

Month: October
Rashi: Vrischika

Month: January
Rashi: Kumbha

Now predict the Rashi for the given month.
Respond with only the Rashi name
'''

### Code Generated:

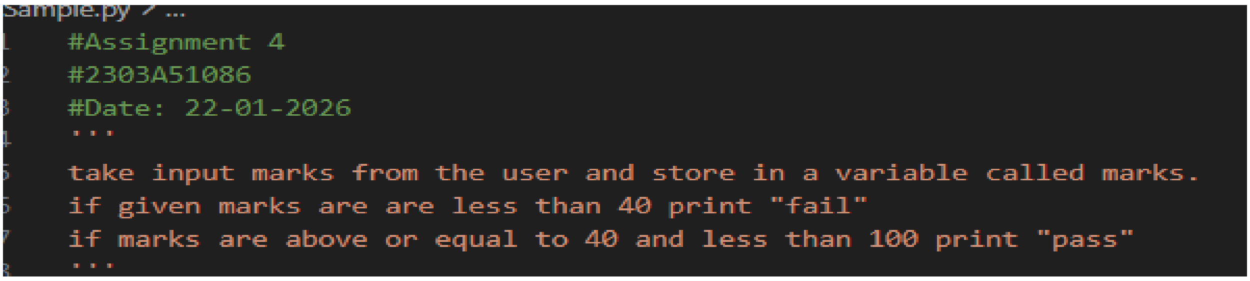
```
month = input("Enter month: ")
month = month.lower()
if month == "january":
   print("Kumbha")
elif month == "february":
   print("Meena")
elif month == "march":
   print("Mesha")
elif month == "april":
   print("Vrishabha")
elif month == "may":
   print("Mithuna")
elif month == "june":
   print("Karka")
```

```
elif month == "july":
    print("Simha")
elif month == "august":
    print("Kanya")
elif month == "september":
    print("Tula")
elif month == "october":
    print("Vrischika")
elif month == "november":
    print("Dhanu")
elif month == "december":
    print("Makara")
else:
    print("Invalid month")
```

## Question 5:

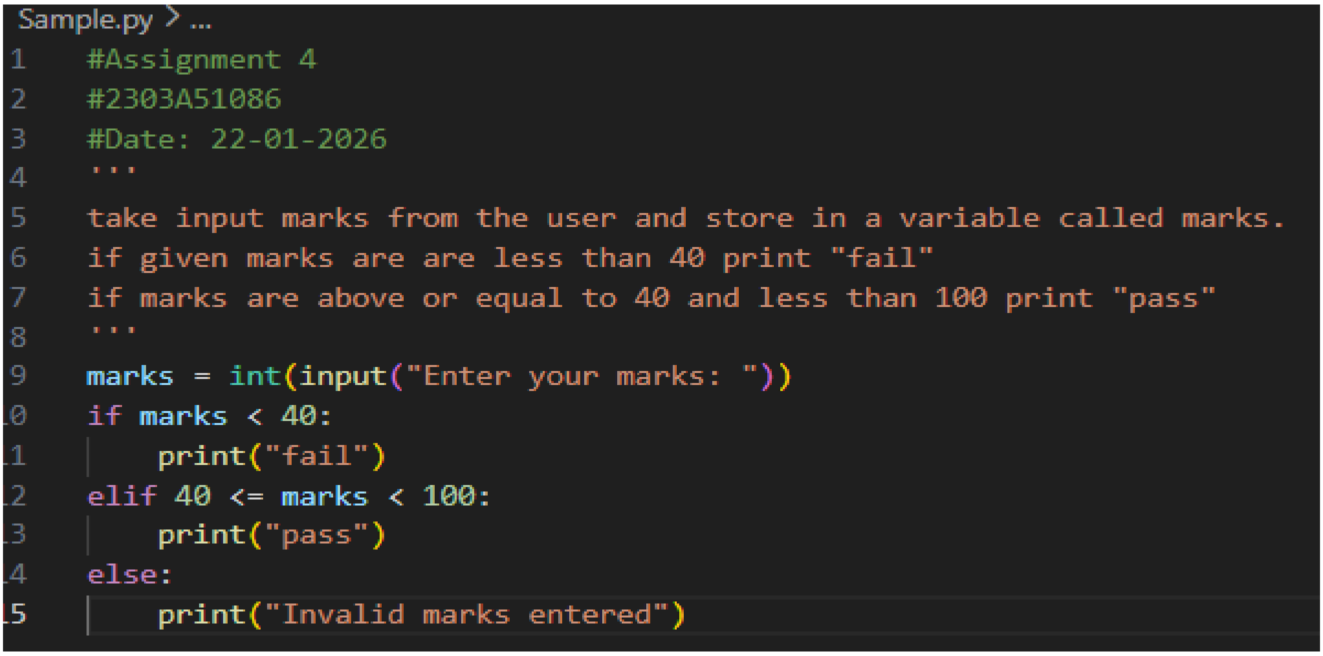
Result Analysis Based on Marks

**Prompt:**



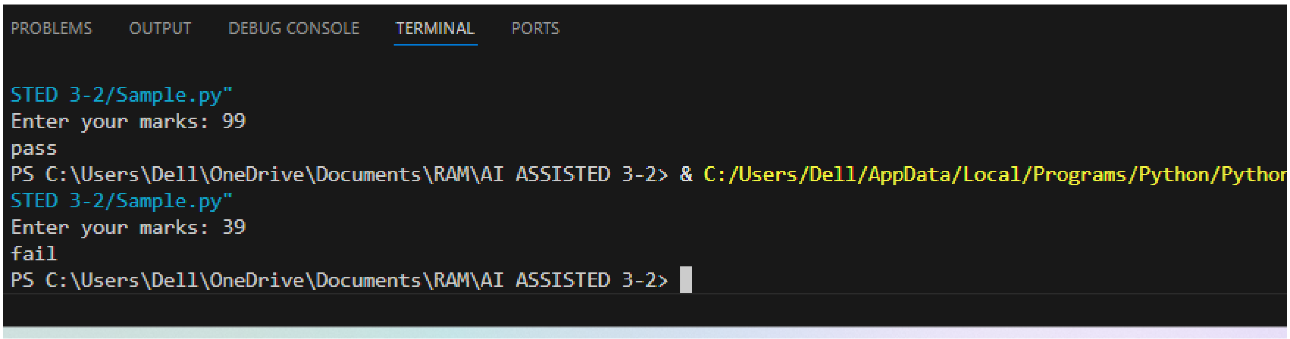
```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
take input marks from the user and store in a variable called marks.
if given marks are are less than 40 print "fail"
if marks are above or equal to 40 and less than 100 print "pass"
'''
```

**Code given by copilot :**



```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
take input marks from the user and store in a variable called marks.
if given marks are are less than 40 print "fail"
if marks are above or equal to 40 and less than 100 print "pass"
'''
marks = int(input("Enter your marks: "))
if marks < 40:
    print("fail")
elif 40 <= marks < 100:
    print("pass")
else:
    print("Invalid marks entered")
```

- **Sample input/output screenshots**



```
STED 3-2/Sample.py"
Enter your marks: 99
pass
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> & C:/Users/Dell/AppData/Local/Programs/Python/Pytho
STED 3-2/Sample.py"
Enter your marks: 39
fail
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2>
```

**Prompt:**

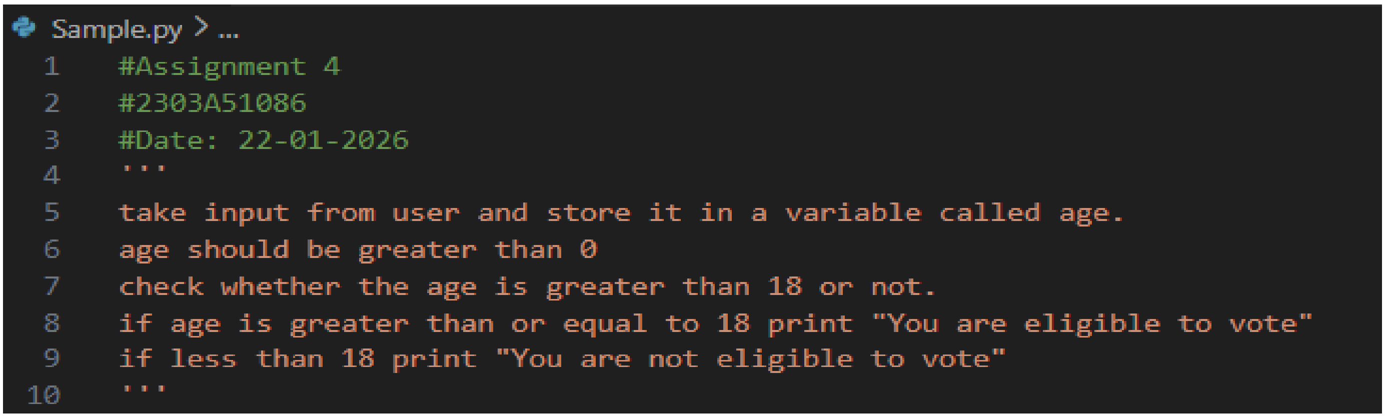
```
'''
take input marks from the user and store in a variable called marks.
if given marks are are less than 40 print "fail"
if marks are above or equal to 40 and less than 100 print "pass"
'''
```

**Code Generated:**

```
marks = int(input("Enter your marks: "))
if marks < 40:
   print("fail")
elif 40 <= marks < 100:
   print("pass")
else:
   print("Invalid marks entered")
```

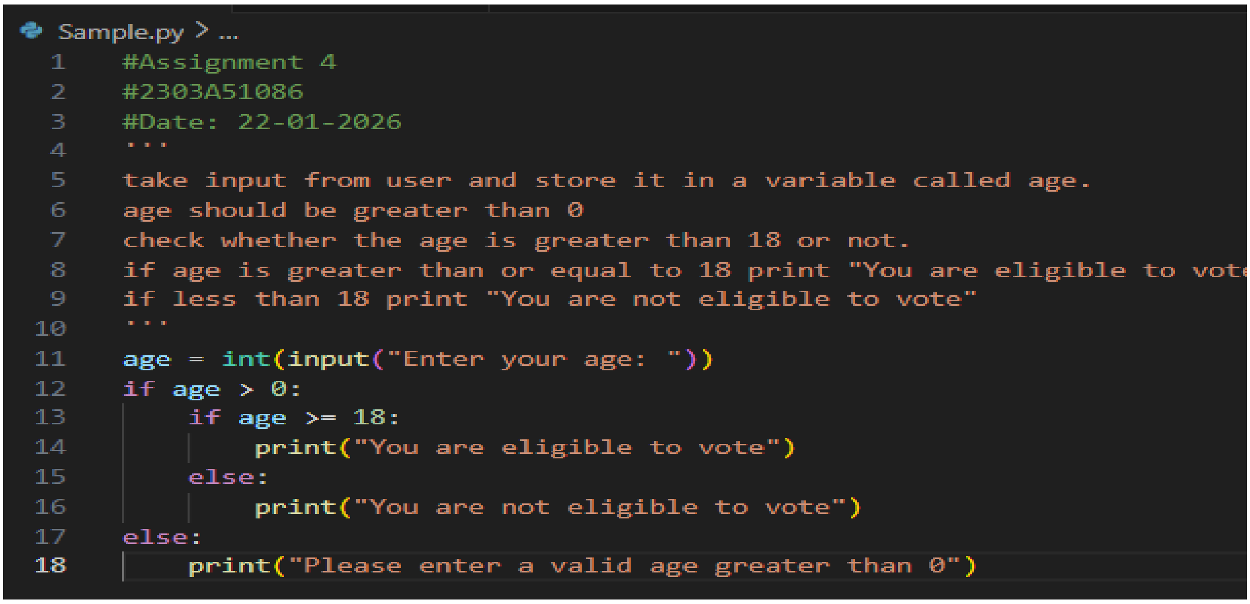
## **Question 6:** Voting Eligibility Check (Chain-of-Thought Prompting)

**Prompt:**



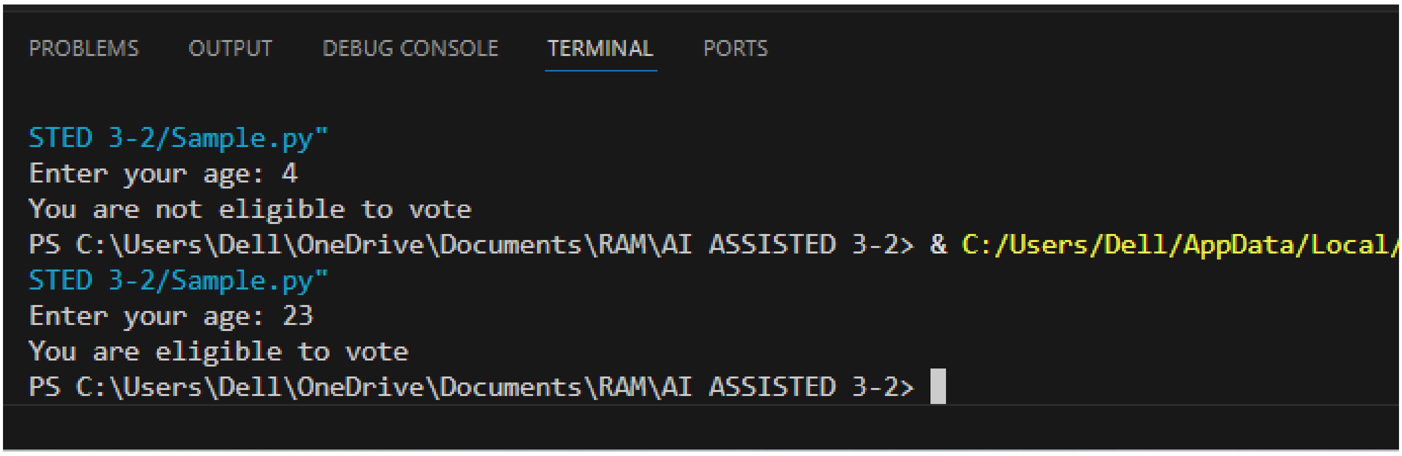
```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
take input from user and store it in a variable called age.
age should be greater than 0
check whether the age is greater than 18 or not.
if age is greater than or equal to 18 print "You are eligible to vote"
if less than 18 print "You are not eligible to vote"
'''
```

**Code given by copilot :**



```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
take input from user and store it in a variable called age.
age should be greater than 0
check whether the age is greater than 18 or not.
if age is greater than or equal to 18 print "You are eligible to vote
if less than 18 print "You are not eligible to vote"
'''
age = int(input("Enter your age: "))
if age > 0:
    if age >= 18:
        print("You are eligible to vote")
    else:
        print("You are not eligible to vote")
else:
    print("Please enter a valid age greater than 0")
```

- **Sample input/output screenshots**



```
STED 3-2/Sample.py"
Enter your age: 4
You are not eligible to vote
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> & C:/Users/Dell/AppData/Local/
STED 3-2/Sample.py"
Enter your age: 23
You are eligible to vote
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> 
```

**Prompt:**

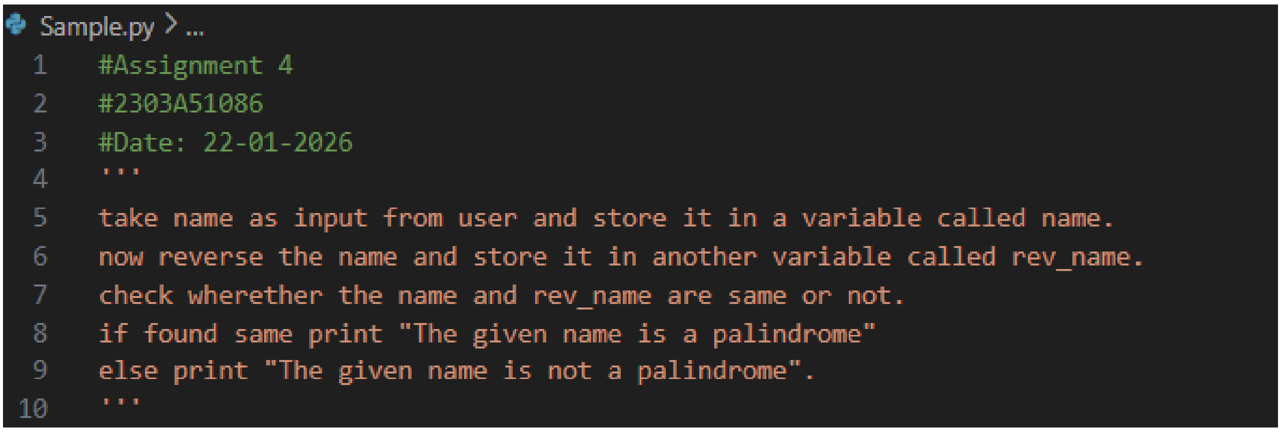
```
'''
take input from user and store it in a variable called age.
age should be greater than 0
check whether the age is greater than 18 or not.
if age is greater than or equal to 18 print "You are eligible to vote"
if less than 18 print "You are not eligible to vote"
'''
```

**Code Generated:**

```
age = int(input("Enter your age: "))
if age > 0:
    if age >= 18:
        print("You are eligible to vote")
    else:
        print("You are not eligible to vote")
else:
    print("Please enter a valid age greater than 0")
```

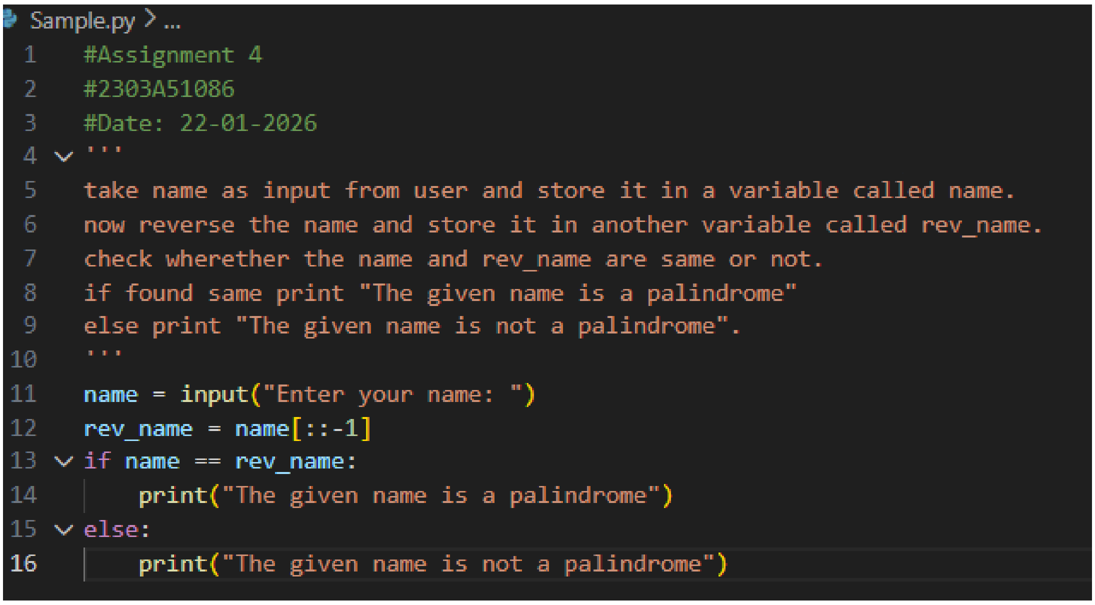
## Question 7: Prompt Chaining (String Processing – Palindrome Names)

**Prompt:**



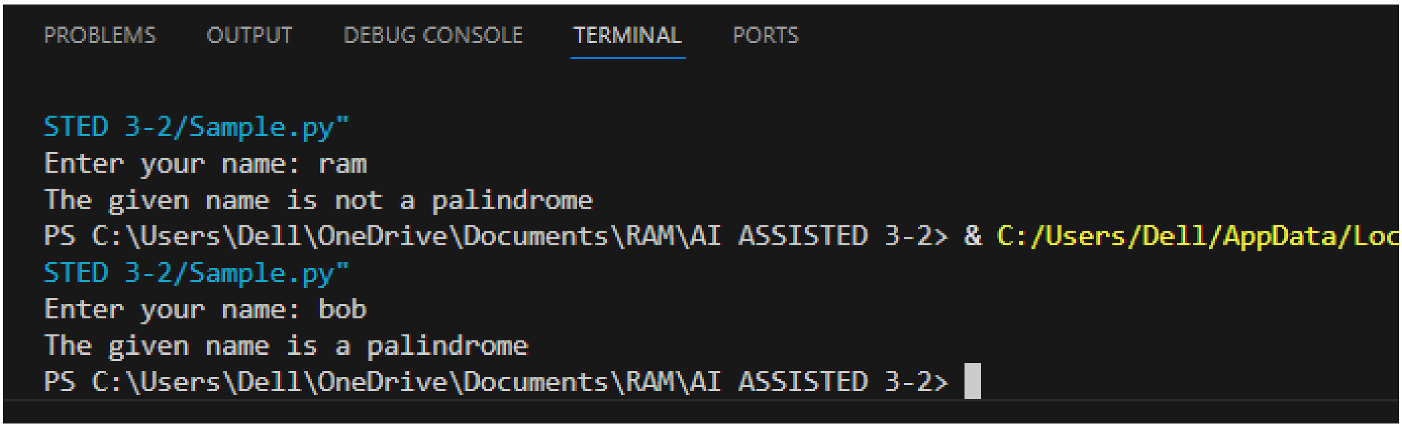
```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
take name as input from user and store it in a variable called name.
now reverse the name and store it in another variable called rev_name.
check wherether the name and rev_name are same or not.
if found same print "The given name is a palindrome"
else print "The given name is not a palindrome".
'''
```

**Code given by copilot :**



```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
take name as input from user and store it in a variable called name.
now reverse the name and store it in another variable called rev_name.
check wherether the name and rev_name are same or not.
if found same print "The given name is a palindrome"
else print "The given name is not a palindrome".
'''
name = input("Enter your name: ")
rev_name = name[::-1]
if name == rev_name:
    print("The given name is a palindrome")
else:
    print("The given name is not a palindrome")
```

- **Sample input/output screenshots**



```
STED 3-2/Sample.py"
Enter your name: ram
The given name is not a palindrome
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> & C:/Users/Dell/AppData/Loc
STED 3-2/Sample.py"
Enter your name: bob
The given name is a palindrome
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2>
```

**Prompt:**

'''

take name as input from user and store it in a variable called name.
now reverse the name and store it in another variable called rev_name.
check wherether the name and rev_name are same or not.
if found same print "The given name is a palindrome"
else print "The given name is not a palindrome".
'''

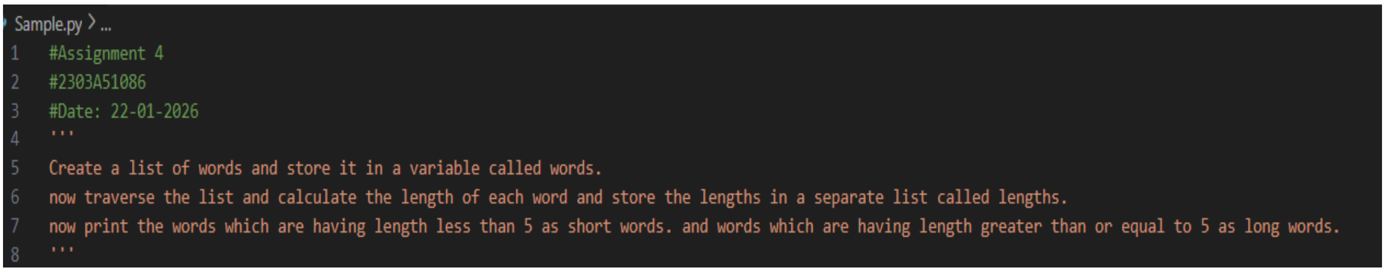
**Code Generated:**

```
name = input("Enter your name: ")
rev_name = name[::-1]
if name == rev_name:
   print("The given name is a palindrome")
```

```
else:
    print("The given name is not a palindrome")
```

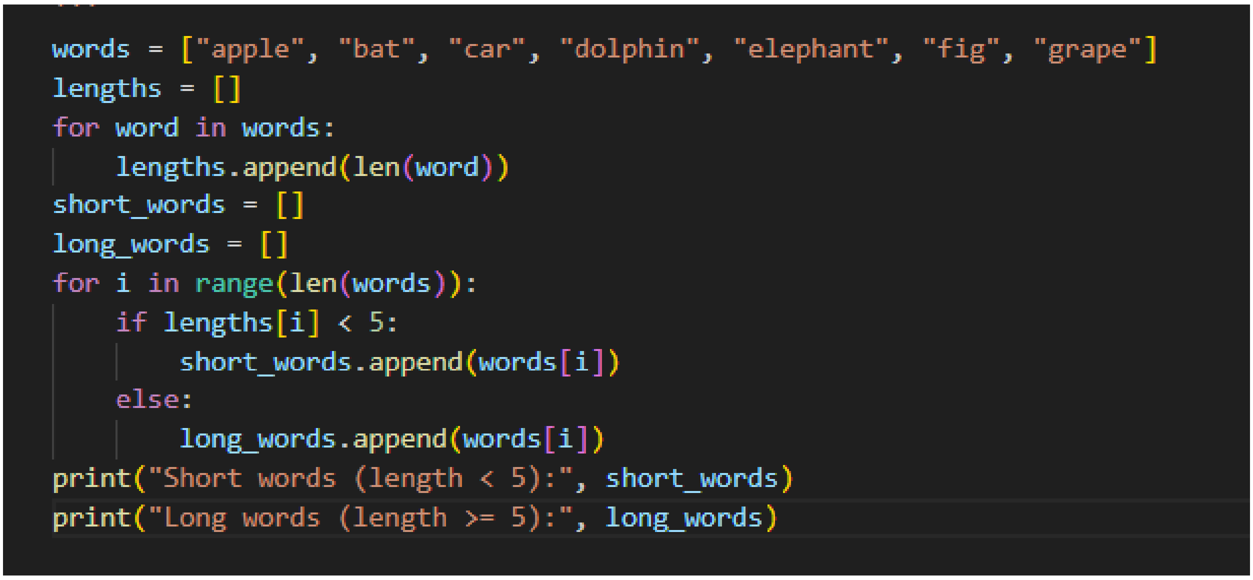
## **Question 8:** Prompt Chaining (String Processing – Word Length Analysis)

**Prompt:**



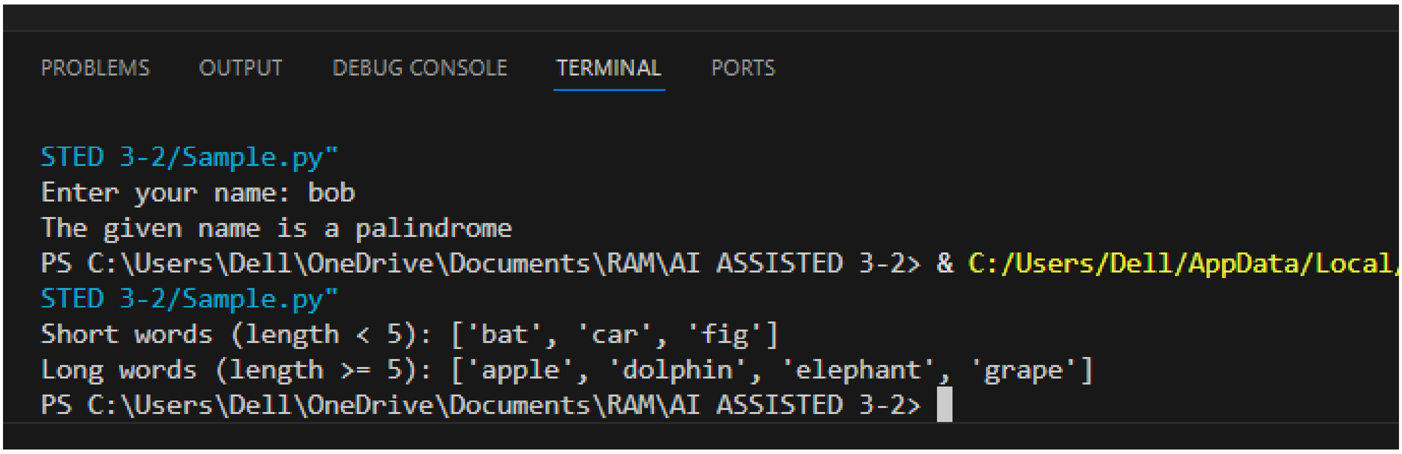
```
#Assignment 4
#2303A51086
#Date: 22-01-2026
'''
Create a list of words and store it in a variable called words.
now traverse the list and calculate the length of each word and store the lengths in a separate list called lengths.
now print the words which are having length less than 5 as short words. and words which are having length greater than or equal to 5 as long words.
'''
```

**Code given by copilot :**



```
words = ["apple", "bat", "car", "dolphin", "elephant", "fig", "grape"]
lengths = []
for word in words:
    lengths.append(len(word))
short_words = []
long_words = []
for i in range(len(words)):
    if lengths[i] < 5:
        short_words.append(words[i])
    else:
        long_words.append(words[i])
print("Short words (length < 5):", short_words)
print("Long words (length >= 5):", long_words)
```

- **Sample input/output screenshots**



```
STED 3-2/Sample.py"
Enter your name: bob
The given name is a palindrome
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2> & C:/Users/Dell/AppData/Local
STED 3-2/Sample.py"
Short words (length < 5): ['bat', 'car', 'fig']
Long words (length >= 5): ['apple', 'dolphin', 'elephant', 'grape']
PS C:\Users\Dell\OneDrive\Documents\RAM\AI ASSISTED 3-2>
```

**Prompt:**

'''
Create a list of words and store it in a variable called words.
now traverse the list and calculate the length of each word and store the lengths in a separate list called lengths.
now print the words which are having length less than 5 as short words. and words which are having length greater than or equal to 5 as long words.
'''

**Code Generated:**

```
words = ["apple", "bat", "car", "dolphin", "elephant", "fig", "grape"]
lengths = []
for word in words:
    lengths.append(len(word))
short_words = []
long_words = []
for i in range(len(words)):
    if lengths[i] < 5:
        short_words.append(words[i])
    else:
        long_words.append(words[i])
print("Short words (length < 5):", short_words)
print("Long words (length >= 5):", long_words)
```