AI Assissted Coding

LAB ASSIGNMENT – 4

Name : G.Abhiram

HT No : 2303A51087

Batch : 02

1) **Zero – Shot Prompting**

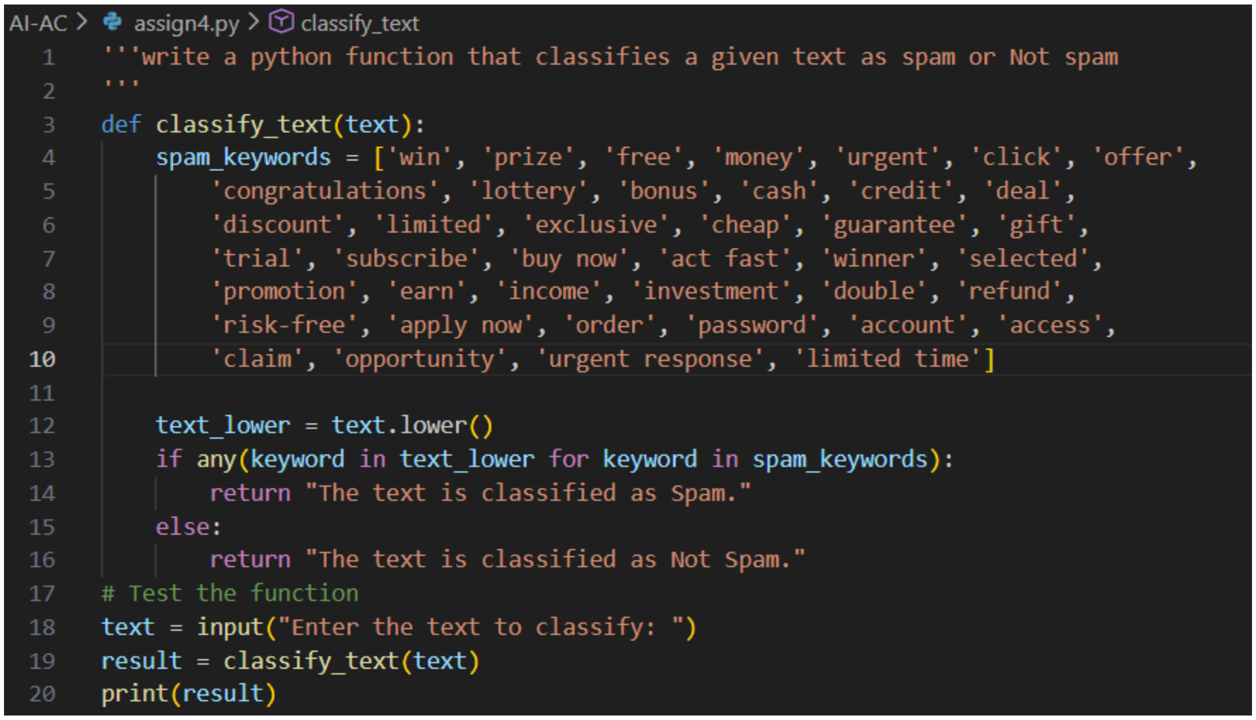
   Task:
   Write a Python function that classifies a given text as Spam or Not Spam using zero-shot prompting.

   Prompt :

   write a python function that classifies a given text as spam or Not spam

   Code :

```
def classify_text(text):
    spam_keywords = ['win', 'prize', 'free', 'money', 'urgent', 'click', 'offer',
        'congratulations', 'lottery', 'bonus', 'cash', 'credit', 'deal',
        'discount', 'limited', 'exclusive', 'cheap', 'guarantee', 'gift',
        'trial', 'subscribe', 'buy now', 'act fast', 'winner', 'selected',
        'promotion', 'earn', 'income', 'investment', 'double', 'refund',
        'risk-free', 'apply now', 'order', 'password', 'account', 'access',
        'claim', 'opportunity', 'urgent response', 'limited time']
    text_lower = text.lower()
    if any(keyword in text_lower for keyword in spam_keywords):
        return "The text is classified as Spam."
    else:
        return "The text is classified as Not Spam."
# Test the function
text = input("Enter the text to classify: ")
result = classify_text(text)
print(result)
```

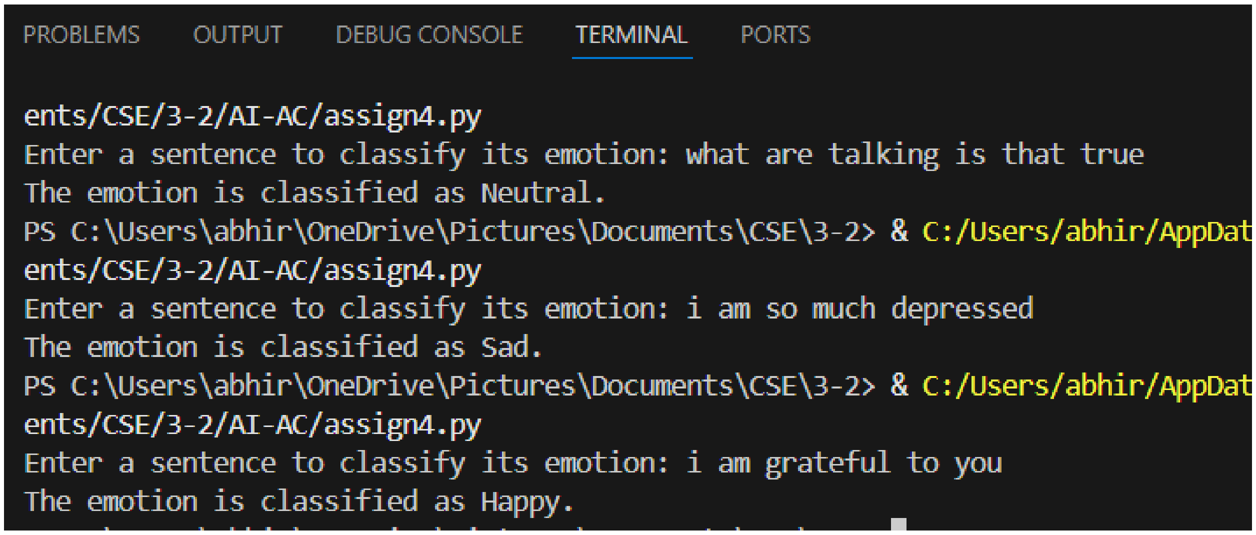


AI-AC > assign4.py > classify_text

```
'''write a python function that classifies a given text as spam or Not spam
'''
def classify_text(text):
    spam_keywords = ['win', 'prize', 'free', 'money', 'urgent', 'click', 'offer',
        'congratulations', 'lottery', 'bonus', 'cash', 'credit', 'deal',
        'discount', 'limited', 'exclusive', 'cheap', 'guarantee', 'gift',
        'trial', 'subscribe', 'buy now', 'act fast', 'winner', 'selected',
        'promotion', 'earn', 'income', 'investment', 'double', 'refund',
        'risk-free', 'apply now', 'order', 'password', 'account', 'access',
        'claim', 'opportunity', 'urgent response', 'limited time']

    text_lower = text.lower()
    if any(keyword in text_lower for keyword in spam_keywords):
        return "The text is classified as Spam."
    else:
        return "The text is classified as Not Spam."
# Test the function
text = input("Enter the text to classify: ")
result = classify_text(text)
print(result)
```

Output :



```
ents/CSE/3-2/AI-AC/assign4.py
Enter a sentence to classify its emotion: what are talking is that true
The emotion is classified as Neutral.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/AppDat
ents/CSE/3-2/AI-AC/assign4.py
Enter a sentence to classify its emotion: i am so much depressed
The emotion is classified as Sad.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/AppDat
ents/CSE/3-2/AI-AC/assign4.py
Enter a sentence to classify its emotion: i am grateful to you
The emotion is classified as Happy.
```

2) One-Shot Prompting

Task :

Write a Python program that detects the emotion of a sentence using one-shot prompting.

Prompt:

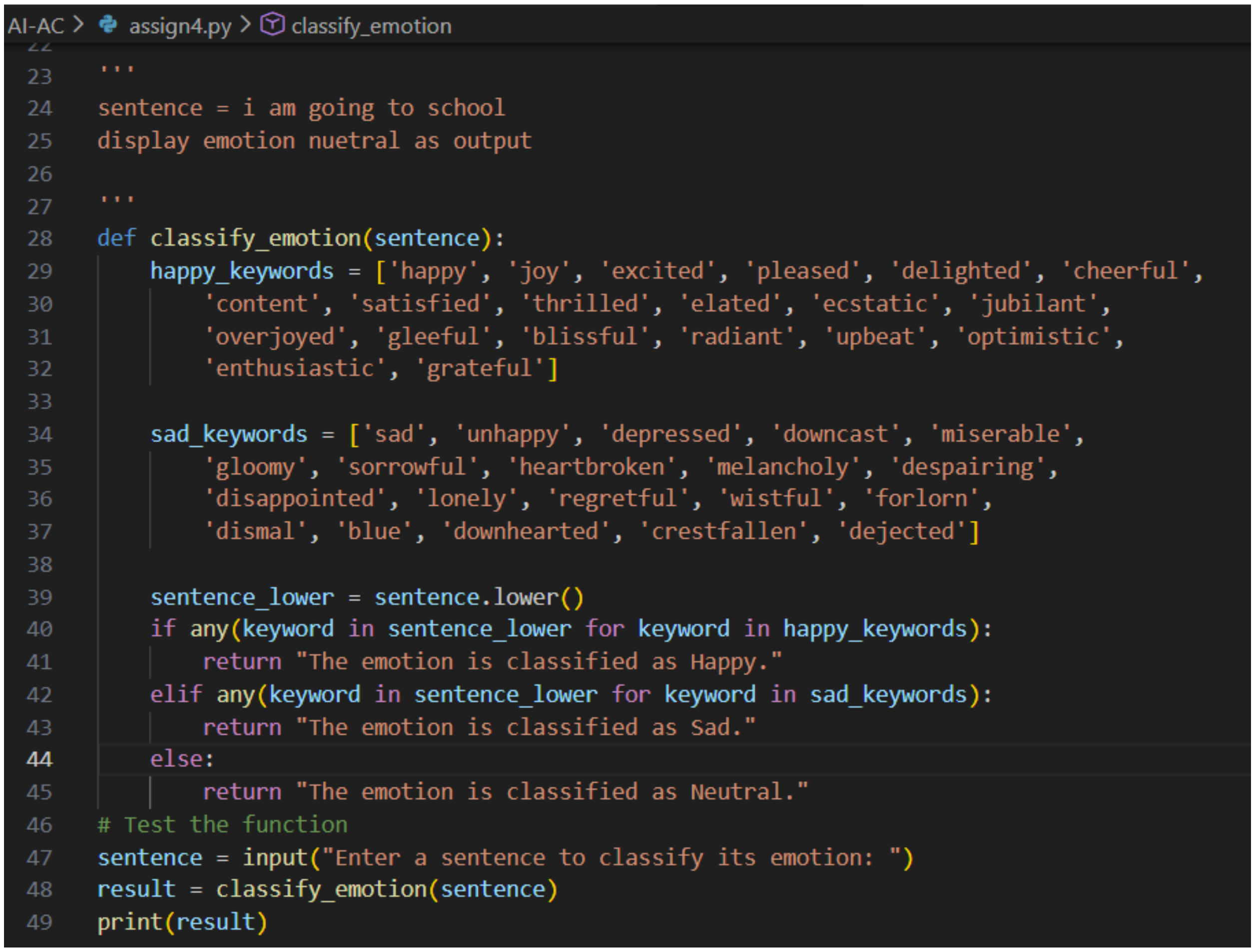
sentence = i am going to school
display emotion nuetral as output

Code :

```
def classify_emotion(sentence):
    happy_keywords = ['happy', 'joy', 'excited', 'pleased', 'delighted',
'cheerful',
        'content', 'satisfied', 'thrilled', 'elated', 'ecstatic', 'jubilant',
        'overjoyed', 'gleeful', 'blissful', 'radiant', 'upbeat', 'optimistic',
        'enthusiastic', 'grateful']

    sad_keywords = ['sad', 'unhappy', 'depressed', 'downcast', 'miserable',
        'gloomy', 'sorrowful', 'heartbroken', 'melancholy', 'despairing',
        'disappointed', 'lonely', 'regretful', 'wistful', 'forlorn',
        'dismal', 'blue', 'downhearted', 'crestfallen', 'dejected']

    sentence_lower = sentence.lower()
    if any(keyword in sentence_lower for keyword in happy_keywords):
        return "The emotion is classified as Happy."
    elif any(keyword in sentence_lower for keyword in sad_keywords):
        return "The emotion is classified as Sad."
    else:
        return "The emotion is classified as Neutral."
# Test the function
sentence = input("Enter a sentence to classify its emotion: ")
result = classify_emotion(sentence)
print(result)
```



AI-AC > assign4.py > classify_emotion

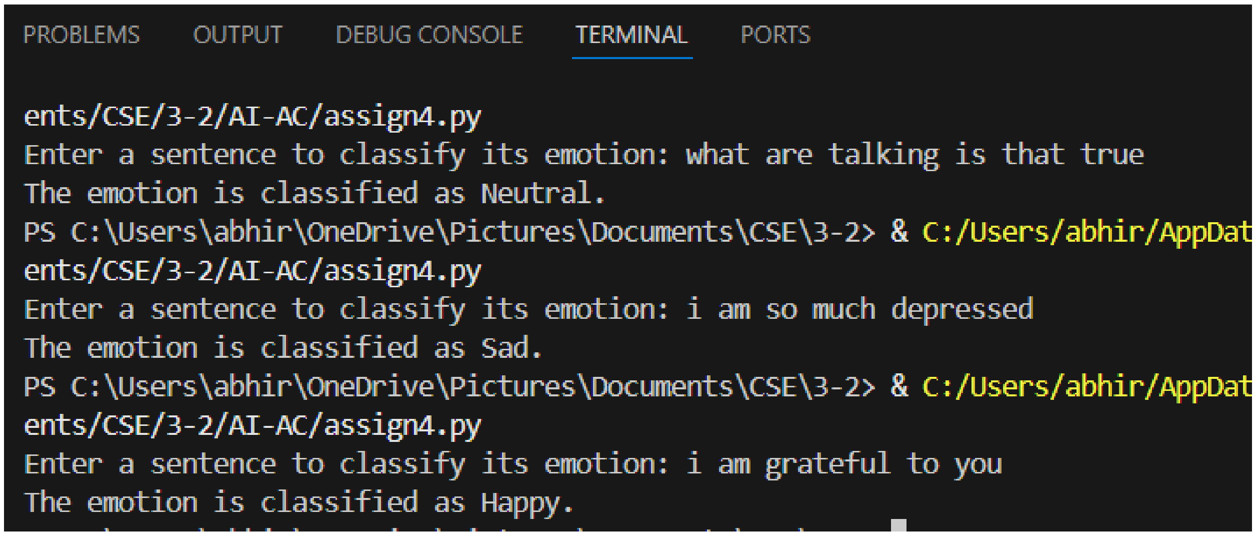
```python
'''
sentence = i am going to school
display emotion nuetral as output

'''
def classify_emotion(sentence):
    happy_keywords = ['happy', 'joy', 'excited', 'pleased', 'delighted', 'cheerful',
        'content', 'satisfied', 'thrilled', 'elated', 'ecstatic', 'jubilant',
        'overjoyed', 'gleeful', 'blissful', 'radiant', 'upbeat', 'optimistic',
        'enthusiastic', 'grateful']

    sad_keywords = ['sad', 'unhappy', 'depressed', 'downcast', 'miserable',
        'gloomy', 'sorrowful', 'heartbroken', 'melancholy', 'despairing',
        'disappointed', 'lonely', 'regretful', 'wistful', 'forlorn',
        'dismal', 'blue', 'downhearted', 'crestfallen', 'dejected']

    sentence_lower = sentence.lower()
    if any(keyword in sentence_lower for keyword in happy_keywords):
        return "The emotion is classified as Happy."
    elif any(keyword in sentence_lower for keyword in sad_keywords):
        return "The emotion is classified as Sad."
    else:
        return "The emotion is classified as Neutral."
# Test the function
sentence = input("Enter a sentence to classify its emotion: ")
result = classify_emotion(sentence)
print(result)
```

Output :



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ents/CSE/3-2/AI-AC/assign4.py
Enter a sentence to classify its emotion: what are talking is that true
The emotion is classified as Neutral.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/AppDat
ents/CSE/3-2/AI-AC/assign4.py
Enter a sentence to classify its emotion: i am so much depressed
The emotion is classified as Sad.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/AppDat
ents/CSE/3-2/AI-AC/assign4.py
Enter a sentence to classify its emotion: i am grateful to you
The emotion is classified as Happy.
```

3) Few – Shot Prompting

Task :

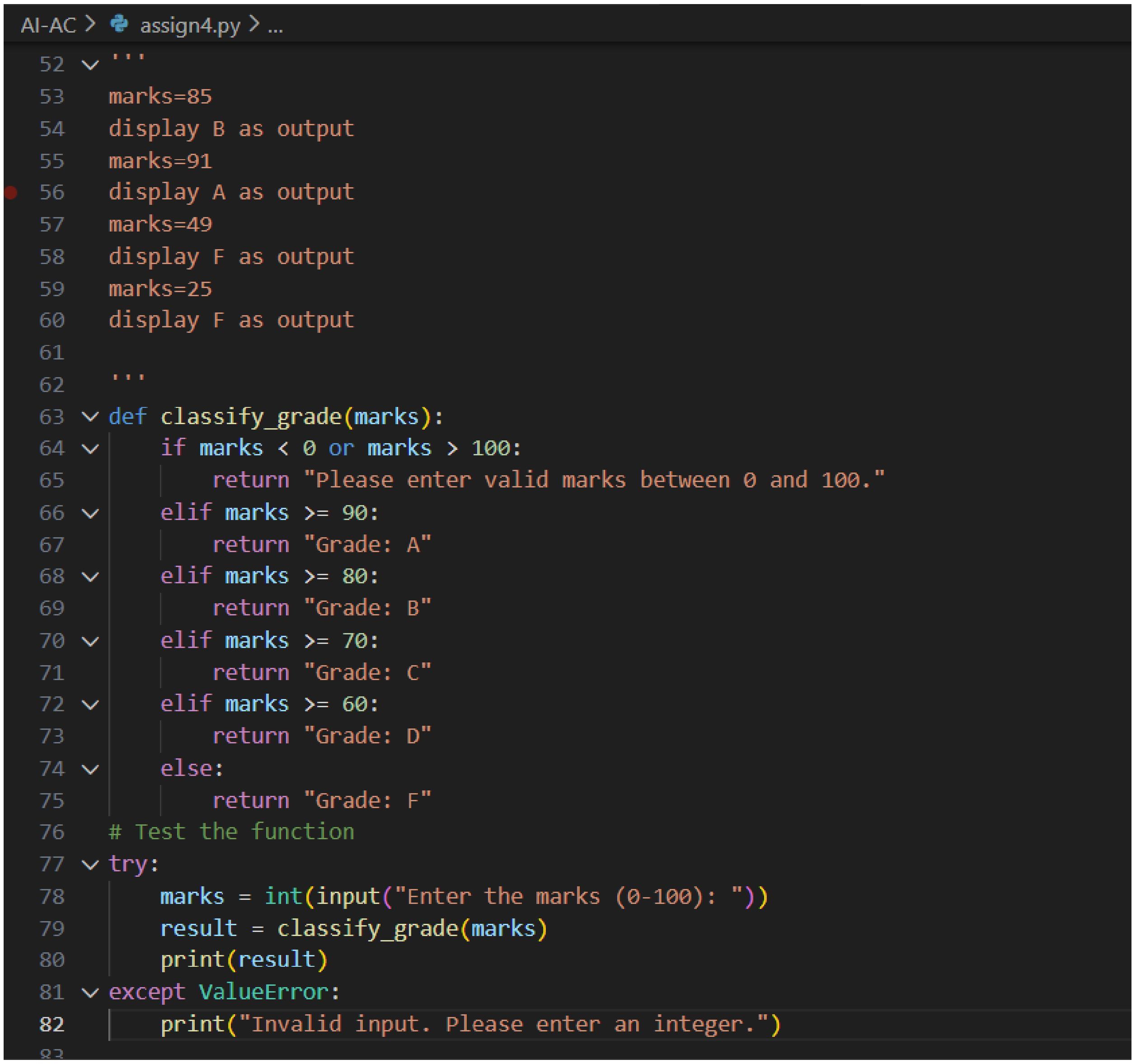
Write a Python program that predicts a student's grade based on marks using few-shot prompting.

Prompt :

marks=85
display B as output
marks=91
display A as output
marks=49
display F as output
marks=25
display F as output

Code :

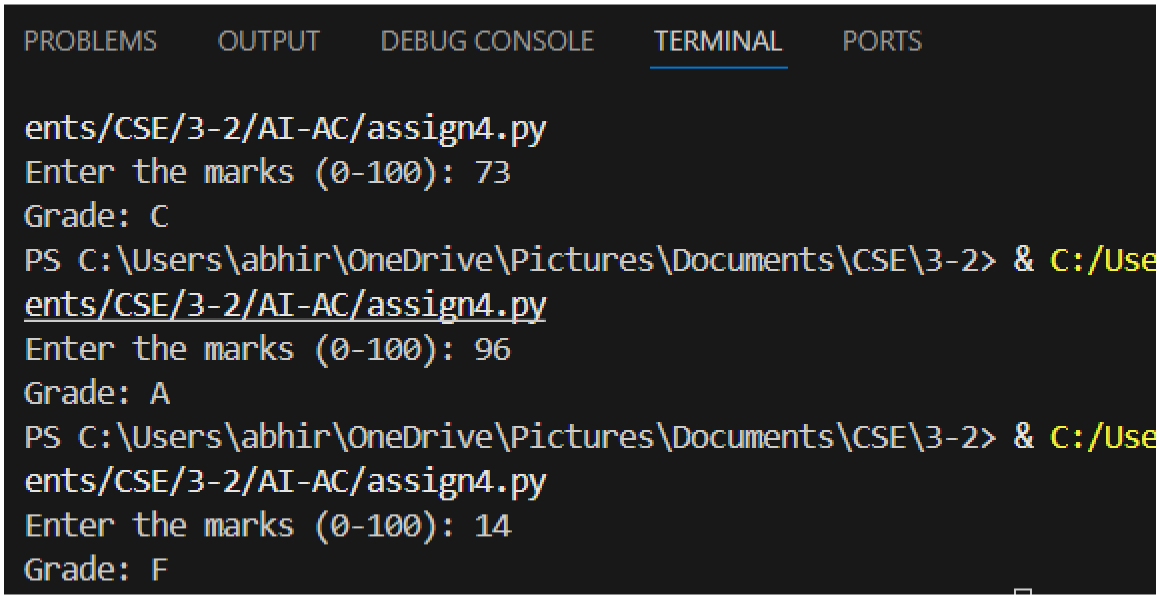
```
def classify_grade(marks):
    if marks < 0 or marks > 100:
        return "Please enter valid marks between 0 and 100."
    elif marks >= 90:
        return "Grade: A"
    elif marks >= 80:
        return "Grade: B"
    elif marks >= 70:
        return "Grade: C"
    elif marks >= 60:
        return "Grade: D"
    else:
        return "Grade: F"
# Test the function
try:
    marks = int(input("Enter the marks (0-100): "))
    result = classify_grade(marks)
    print(result)
except ValueError:
    print("Invalid input. Please enter an integer.")
```


AI-AC > assign4.py > ...

```python
'''
marks=85
display B as output
marks=91
display A as output
marks=49
display F as output
marks=25
display F as output

'''
def classify_grade(marks):
    if marks < 0 or marks > 100:
        return "Please enter valid marks between 0 and 100."
    elif marks >= 90:
        return "Grade: A"
    elif marks >= 80:
        return "Grade: B"
    elif marks >= 70:
        return "Grade: C"
    elif marks >= 60:
        return "Grade: D"
    else:
        return "Grade: F"
# Test the function
try:
    marks = int(input("Enter the marks (0-100): "))
    result = classify_grade(marks)
    print(result)
except ValueError:
    print("Invalid input. Please enter an integer.")
```

Output :



```
ents/CSE/3-2/AI-AC/assign4.py
Enter the marks (0-100): 73
Grade: C
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Use
ents/CSE/3-2/AI-AC/assign4.py
Enter the marks (0-100): 96
Grade: A
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Use
ents/CSE/3-2/AI-AC/assign4.py
Enter the marks (0-100): 14
Grade: F
```

4) Multi Shot Prompting

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.

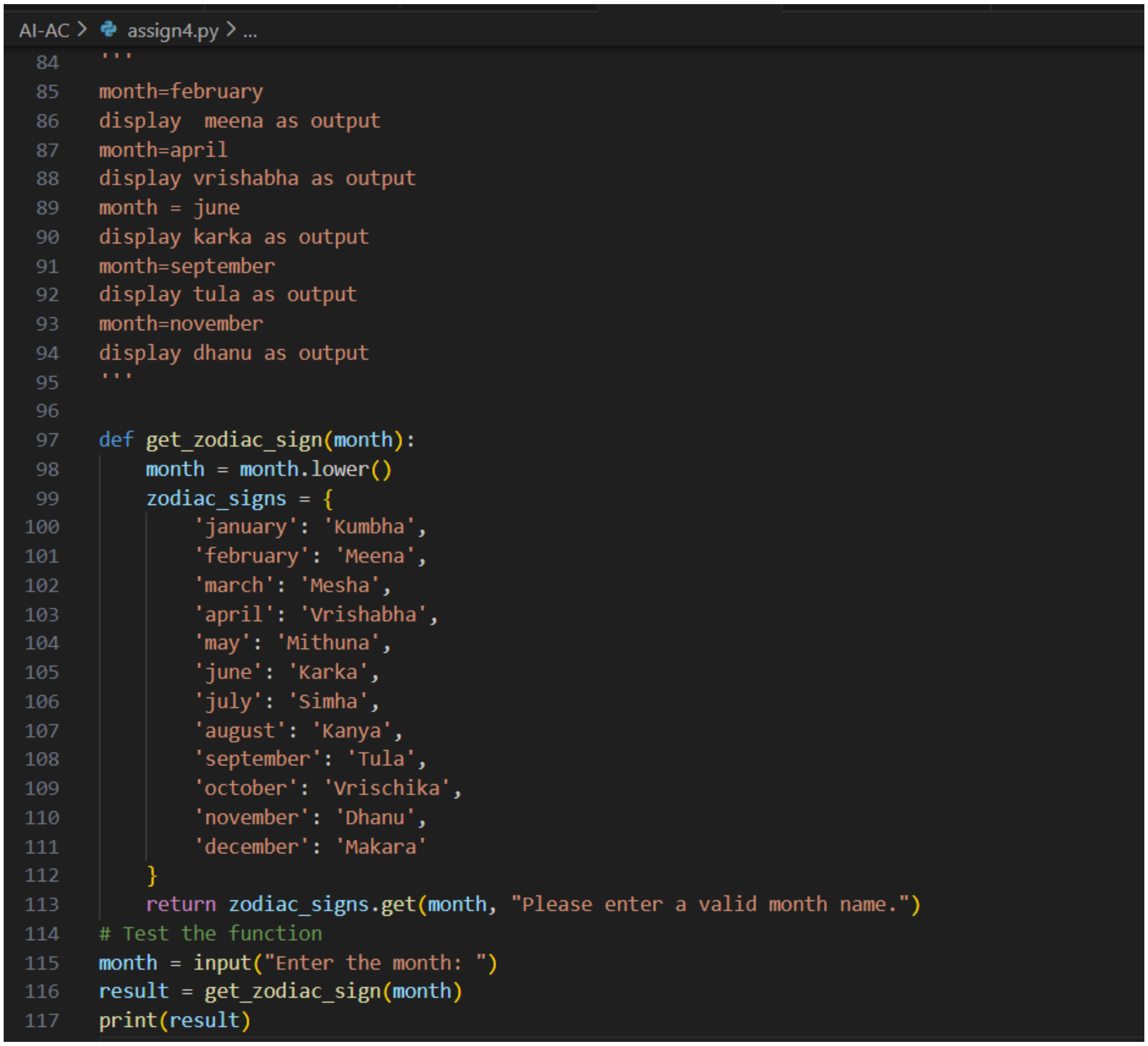
Prompt :

month=february
display meena as output
month=april
display vrishabha as output
month = june
display karka as output
month=september
display tula as output

month=november
display dhanu as output

Code :

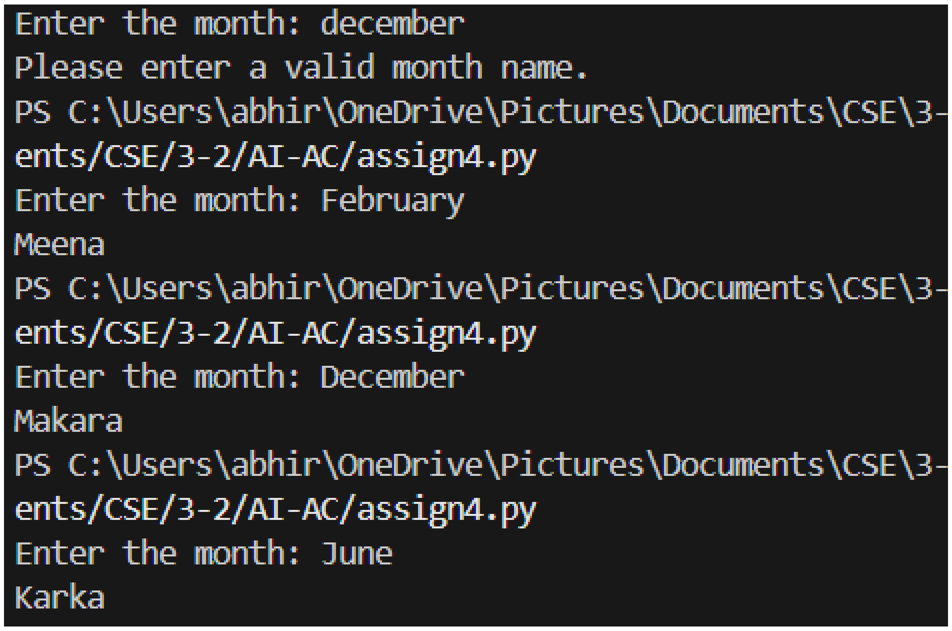
```
def get_zodiac_sign(month):
    month = month.lower()
    zodiac_signs = {
        'january': 'Kumbha',
        'february': 'Meena',
        'march': 'Mesha',
        'april': 'Vrishabha',
        'may': 'Mithuna',
        'june': 'Karka',
        'july': 'Simha',
        'august': 'Kanya',
        'september': 'Tula',
        'october': 'Vrischika',
        'november': 'Dhanu',
        'december': 'Makara'
    }
    return zodiac_signs.get(month, "Please enter a valid month name.")
# Test the function
month = input("Enter the month: ")
result = get_zodiac_sign(month)
print(result)
```


AI-AC > assign4.py > ...

```
'''
month=february
display  meena as output
month=april
display vrishabha as output
month = june
display karka as output
month=september
display tula as output
month=november
display dhanu as output
'''

def get_zodiac_sign(month):
    month = month.lower()
    zodiac_signs = {
        'january': 'Kumbha',
        'february': 'Meena',
        'march': 'Mesha',
        'april': 'Vrishabha',
        'may': 'Mithuna',
        'june': 'Karka',
        'july': 'Simha',
        'august': 'Kanya',
        'september': 'Tula',
        'october': 'Vrischika',
        'november': 'Dhanu',
        'december': 'Makara'
    }
    return zodiac_signs.get(month, "Please enter a valid month name.")
# Test the function
month = input("Enter the month: ")
result = get_zodiac_sign(month)
print(result)
```

Output :


```
Enter the month: december
Please enter a valid month name.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-
ents/CSE/3-2/AI-AC/assign4.py
Enter the month: February
Meena
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-
ents/CSE/3-2/AI-AC/assign4.py
Enter the month: December
Makara
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-
ents/CSE/3-2/AI-AC/assign4.py
Enter the month: June
Karka
```

5) **Voting Eligibility Check**

Task :

Write a Python program that determines whether a person is eligible to vote using Chain-of-Thought (CoT) prompting.

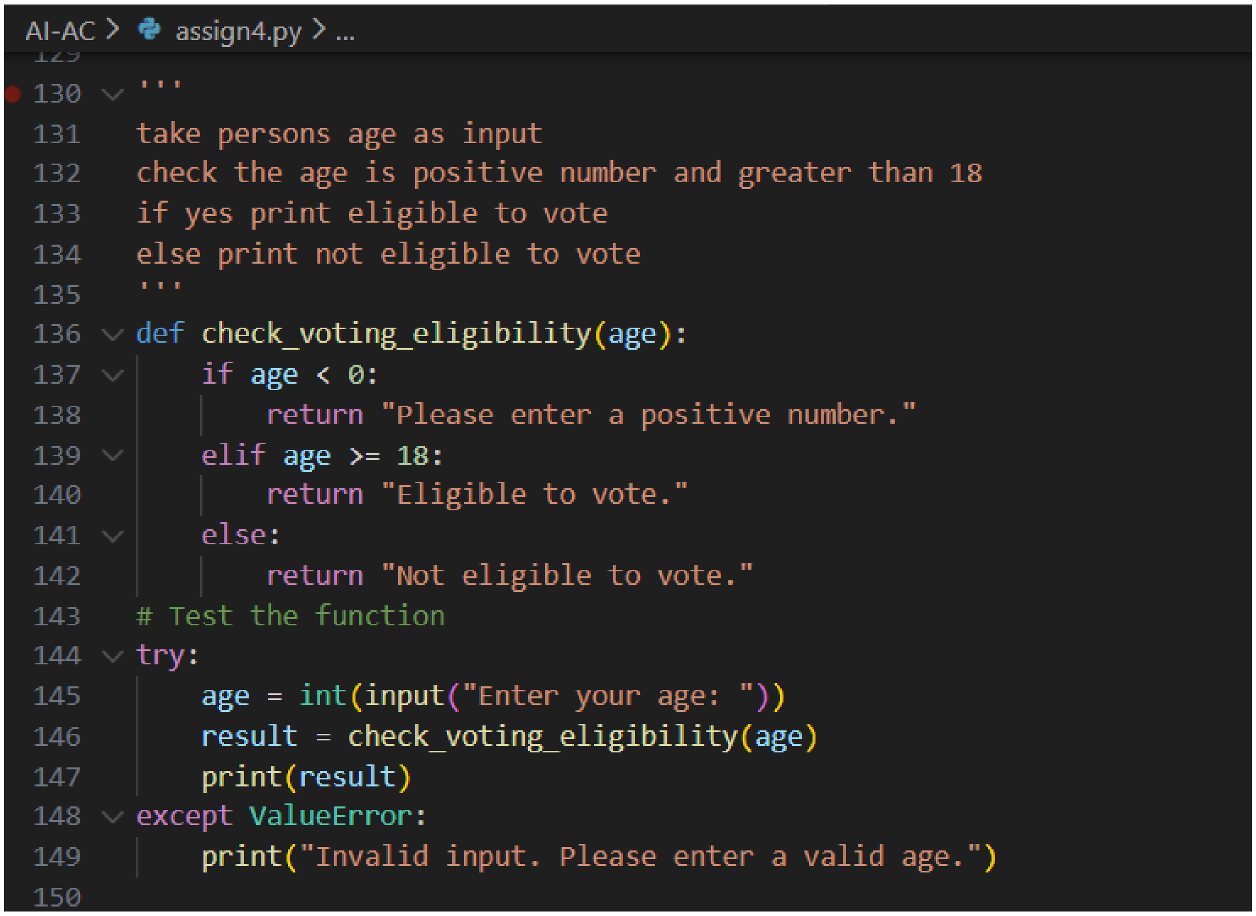
Prompt :

take persons age as input
check the age is positive number and greater than 18
if yes print eligible to vote
else print not eligible to vote

Code :

def check_voting_eligibility(age):

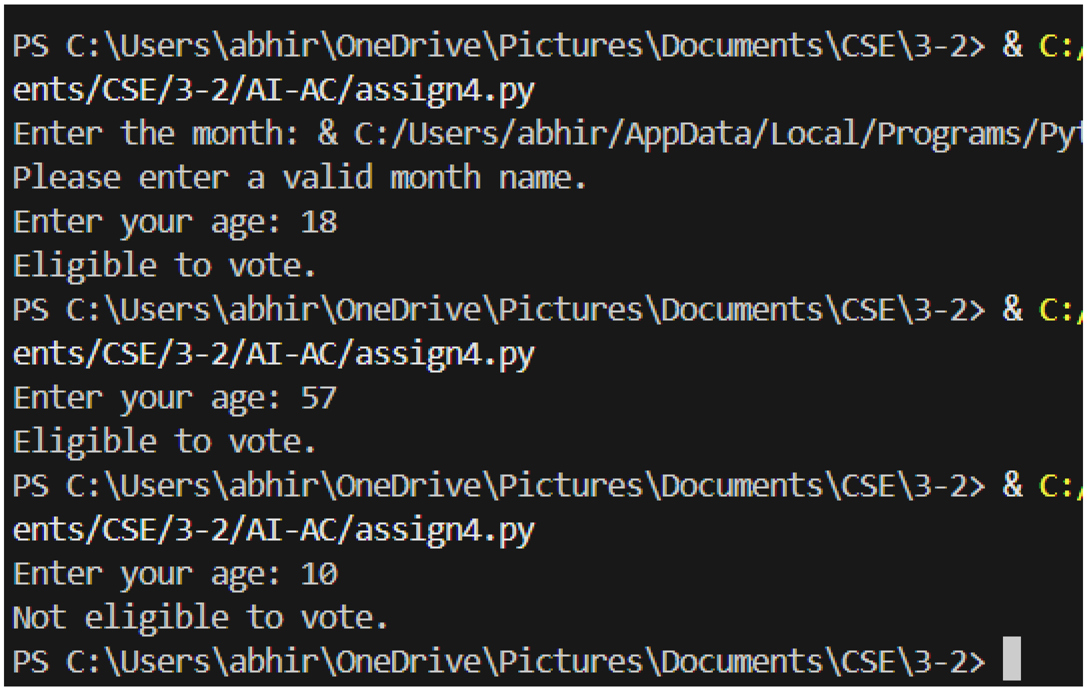
```
    if age < 0:
        return "Please enter a positive number."
    elif age >= 18:
        return "Eligible to vote."
    else:
        return "Not eligible to vote."
# Test the function
try:
    age = int(input("Enter your age: "))
    result = check_voting_eligibility(age)
    print(result)
except ValueError:
    print("Invalid input. Please enter a valid age.")
```



AI-AC > assign4.py > ...

```
'''
take persons age as input
check the age is positive number and greater than 18
if yes print eligible to vote
else print not eligible to vote
'''
def check_voting_eligibility(age):
    if age < 0:
        return "Please enter a positive number."
    elif age >= 18:
        return "Eligible to vote."
    else:
        return "Not eligible to vote."
# Test the function
try:
    age = int(input("Enter your age: "))
    result = check_voting_eligibility(age)
    print(result)
except ValueError:
    print("Invalid input. Please enter a valid age.")
```

Output :



```
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/
ents/CSE/3-2/AI-AC/assign4.py
Enter the month: & C:/Users/abhir/AppData/Local/Programs/Pyt
Please enter a valid month name.
Enter your age: 18
Eligible to vote.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/
ents/CSE/3-2/AI-AC/assign4.py
Enter your age: 57
Eligible to vote.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/
ents/CSE/3-2/AI-AC/assign4.py
Enter your age: 10
Not eligible to vote.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2>
```

6) **Prompt Chaining**

Task :

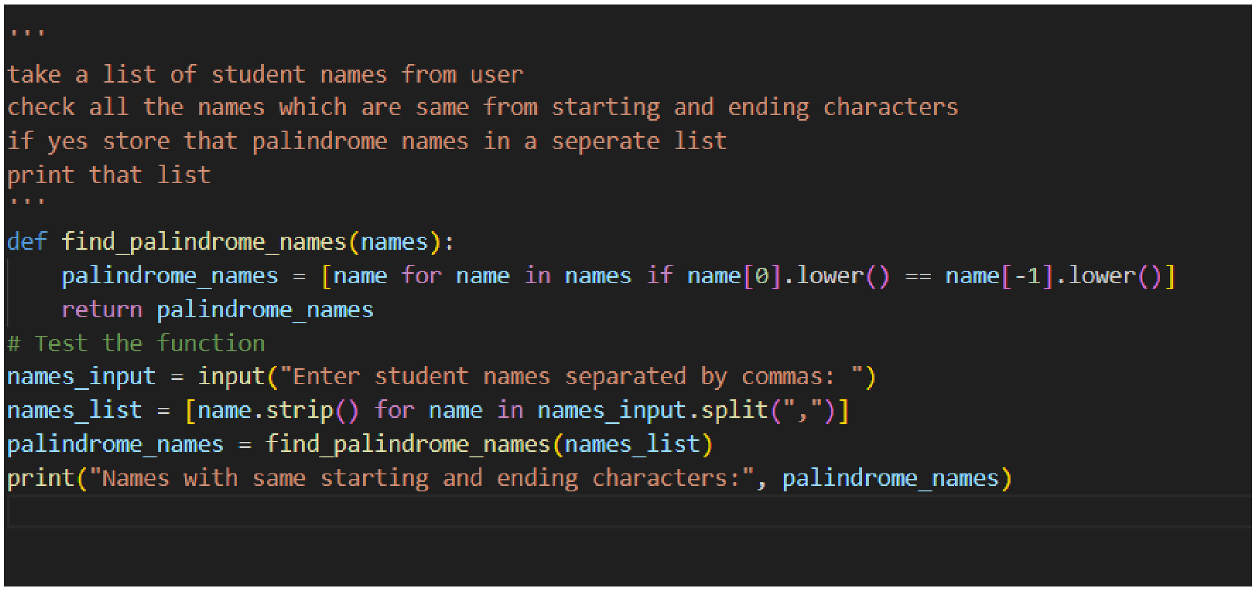
Write a Python program that uses the prompt chaining technique to identify palindrome names from a list of student names.

Prompt :

take a list of student names from user
check all the names which are same from starting and ending characters
if yes store that palindrome names in a seperate list
print that list

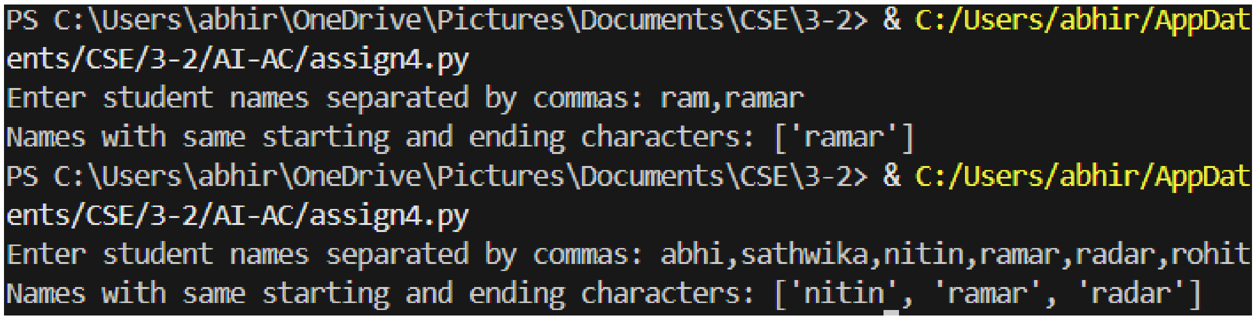
Code :

```
def find_palindrome_names(names):
    palindrome_names = [name for name in names if name[0].lower() ==
name[-1].lower()]
    return palindrome_names
# Test the function
names_input = input("Enter student names separated by commas: ")
names_list = [name.strip() for name in names_input.split(",")]
palindrome_names = find_palindrome_names(names_list)
print("Names with same starting and ending characters:",
palindrome_names)
```



```
'''
take a list of student names from user
check all the names which are same from starting and ending characters
if yes store that palindrome names in a seperate list
print that list
'''
def find_palindrome_names(names):
    palindrome_names = [name for name in names if name[0].lower() == name[-1].lower()]
    return palindrome_names
# Test the function
names_input = input("Enter student names separated by commas: ")
names_list = [name.strip() for name in names_input.split(",")]
palindrome_names = find_palindrome_names(names_list)
print("Names with same starting and ending characters:", palindrome_names)
```

Output :



```
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/AppDat
ents/CSE/3-2/AI-AC/assign4.py
Enter student names separated by commas: ram,ramar
Names with same starting and ending characters: ['ramar']
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/AppDat
ents/CSE/3-2/AI-AC/assign4.py
Enter student names separated by commas: abhi,sathwika,nitin,ramar,radar,rohit
Names with same starting and ending characters: ['nitin', 'ramar', 'radar']
```

7) Prompt Chaining

Task :

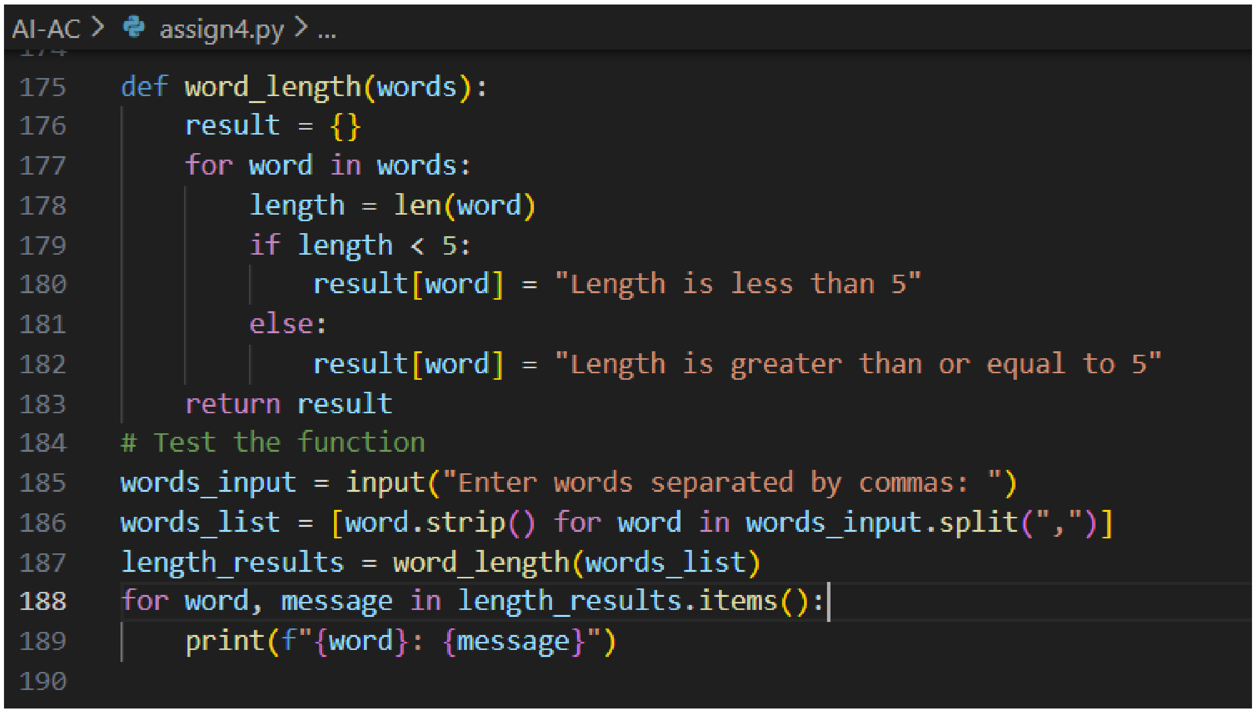
Write a Python program that uses prompt chaining to
analyze a list of words. In the first prompt, generate a list of words.
In the second prompt, traverse the list and calculate the length of
each word. In the third prompt, use the output of the previous step
to determine whether each word is Short (length less than 5) or
Long (length greater than or equal to 5), and display the result for
each word

Prompt :

generate list a words
from list of words calculate the length of each word
display the list of words whether word length is less than 5 or greater than
5 and result of each word

Code :

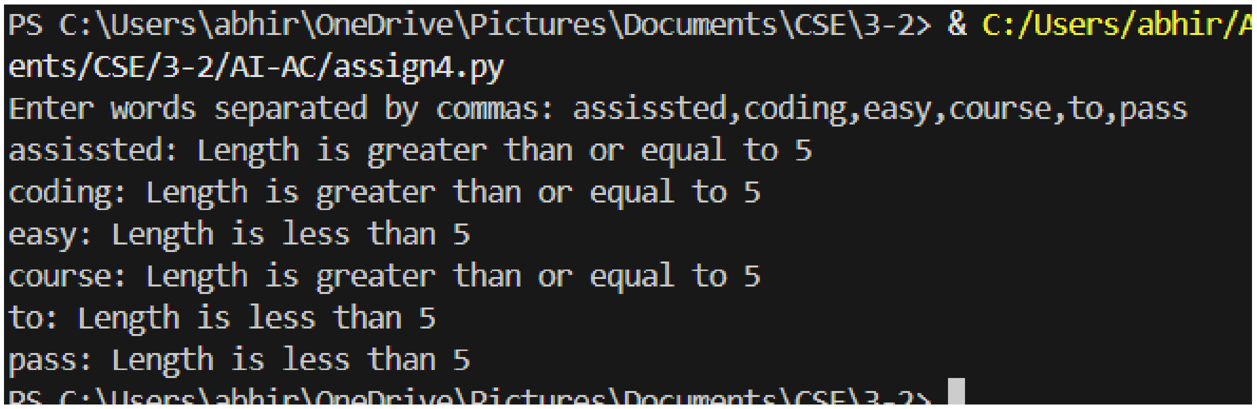
```
def word_length(words):
    result = {}
    for word in words:
        length = len(word)
        if length < 5:
            result[word] = "Length is less than 5"
        else:
            result[word] = "Length is greater than or equal to 5"
    return result
# Test the function
words_input = input("Enter words separated by commas: ")
words_list = [word.strip() for word in words_input.split(",")]
length_results = word_length(words_list)
for word, message in length_results.items():
    print(f"{word}: {message}")
```



AI-AC > assign4.py > ...

```
def word_length(words):
    result = {}
    for word in words:
        length = len(word)
        if length < 5:
            result[word] = "Length is less than 5"
        else:
            result[word] = "Length is greater than or equal to 5"
    return result
# Test the function
words_input = input("Enter words separated by commas: ")
words_list = [word.strip() for word in words_input.split(",")]
length_results = word_length(words_list)
for word, message in length_results.items():
    print(f"{word}: {message}")
```

Output :



```
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/A
ents/CSE/3-2/AI-AC/assign4.py
Enter words separated by commas: assissted,coding,easy,course,to,pass
assissted: Length is greater than or equal to 5
coding: Length is greater than or equal to 5
easy: Length is less than 5
course: Length is greater than or equal to 5
to: Length is less than 5
pass: Length is less than 5
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2>
```