ASSIGNIMNET -4

HALLTICKET NO:2303A10A6

BATCH NO:02

1. 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.

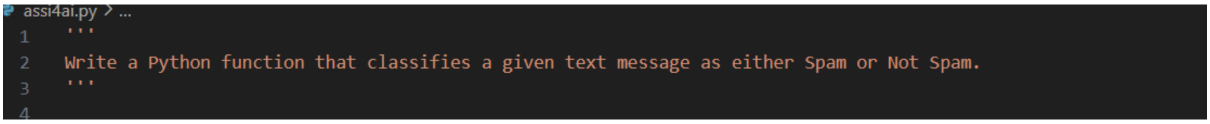
Input:

"Congratulations! You have won a free lottery ticket."

Expected Output:

Spam

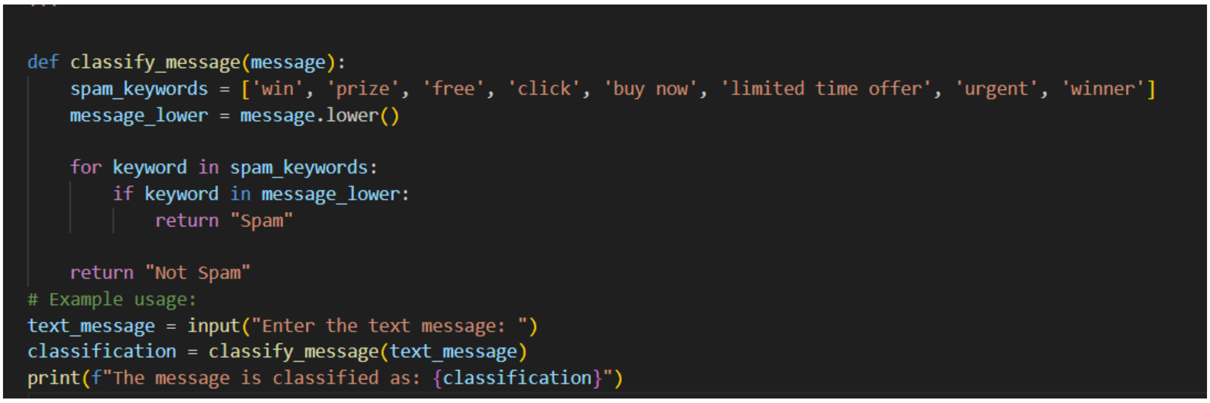
Prompt:



assi4ai.py > ...

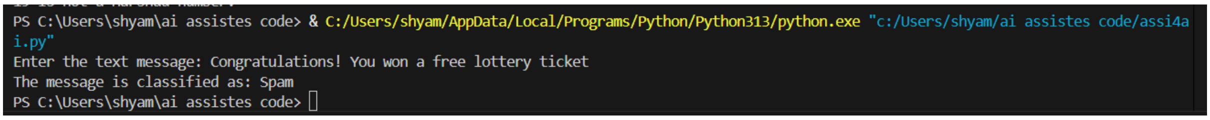
```
'''
Write a Python function that classifies a given text message as either Spam or Not Spam.
'''
```

Code:



```
def classify_message(message):
    spam_keywords = ['win', 'prize', 'free', 'click', 'buy now', 'limited time offer', 'urgent', 'winner']
    message_lower = message.lower()

    for keyword in spam_keywords:
        if keyword in message_lower:
            return "Spam"

    return "Not Spam"
# Example usage:
text_message = input("Enter the text message: ")
classification = classify_message(text_message)
print(f"The message is classified as: {classification}")
```

Output :



```
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai assistes code/assi4a
i.py"
Enter the text message: Congratulations! You won a free lottery ticket
The message is classified as: Spam
PS C:\Users\shyam\ai assistes code> 
```

2. 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’]

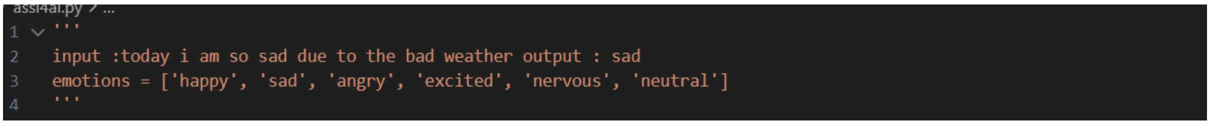
Steps:

1. Provide one labeled example inside the prompt.

2. Take a sentence as input.

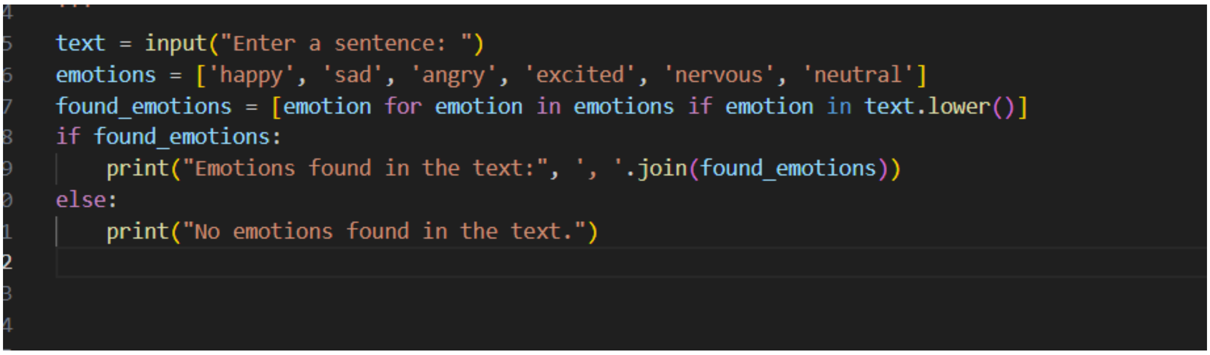
3. Print the predicted emotion

Prompt:



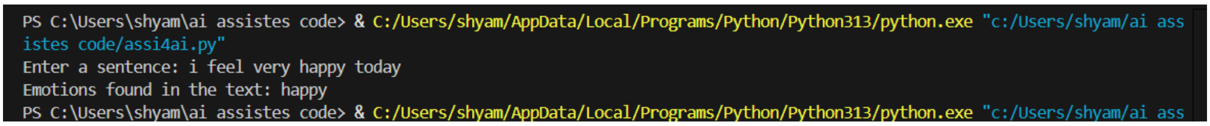
```
assi4ai.py > ...
'''
input :today i am so sad due to the bad weather output : sad
emotions = ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']
'''
```

Code:



```
text = input("Enter a sentence: ")
emotions = ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']
found_emotions = [emotion for emotion in emotions if emotion in text.lower()]
if found_emotions:
    print("Emotions found in the text:", ', '.join(found_emotions))
else:
    print("No emotions found in the text.")
```

Output:



```
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter a sentence: i feel very happy today
Emotions found in the text: happy
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
```

3. 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.

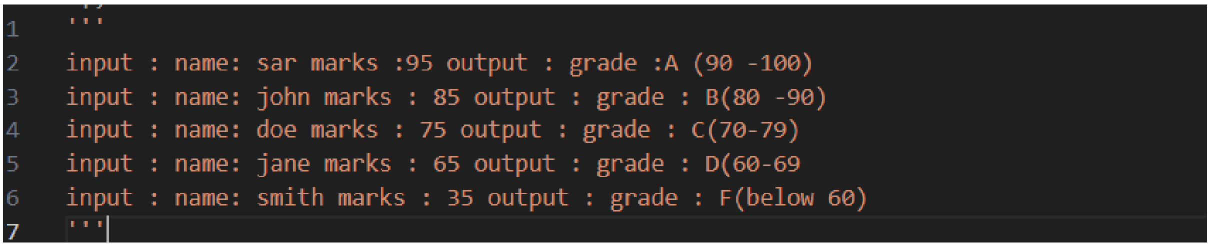
Grades:

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

Grading Criteria (to be inferred from examples):

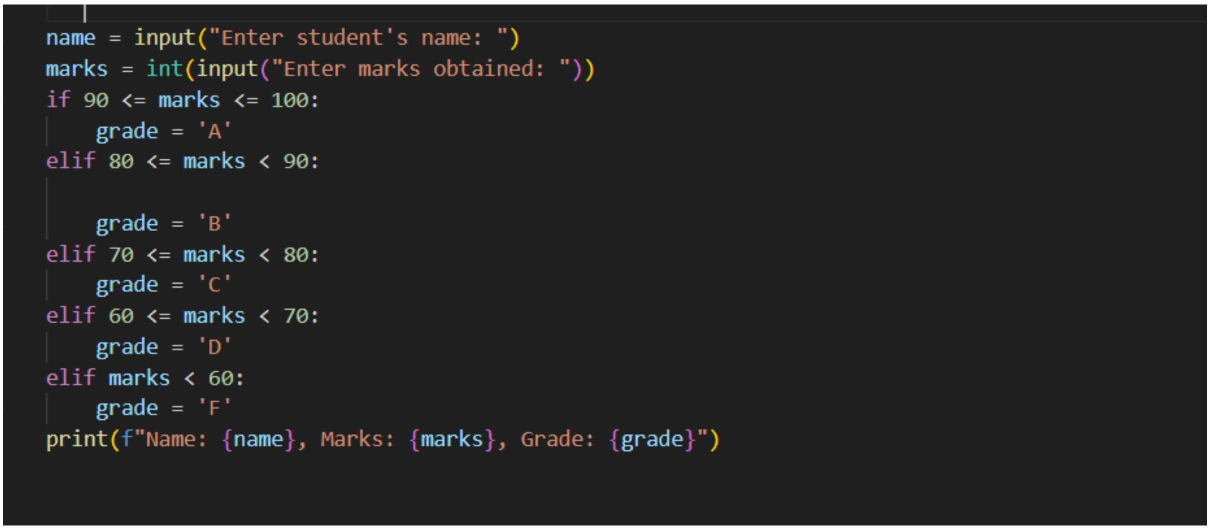
- 90–100 → A
- 80–89 → B
- 70–79 → C
- 60–69 → D
- Below 60 → F

Prompt:



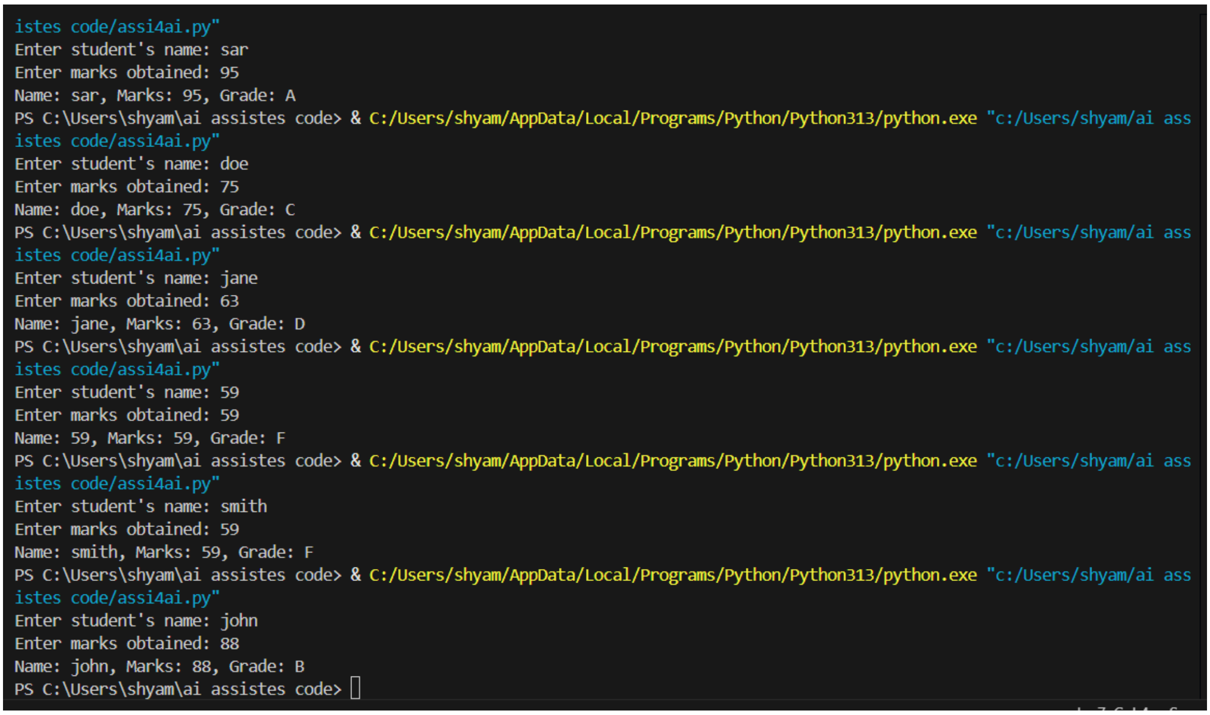
```
'''
input : name: sar marks :95 output : grade :A (90 -100)
input : name: john marks : 85 output : grade : B(80 -90)
input : name: doe marks : 75 output : grade : C(70-79)
input : name: jane marks : 65 output : grade : D(60-69
input : name: smith marks : 35 output : grade : F(below 60)
'''
```

Code:



```
name = input("Enter student's name: ")
marks = int(input("Enter marks obtained: "))
if 90 <= marks <= 100:
    grade = 'A'
elif 80 <= marks < 90:

    grade = 'B'
elif 70 <= marks < 80:
    grade = 'C'
elif 60 <= marks < 70:
    grade = 'D'
elif marks < 60:
    grade = 'F'
print(f"Name: {name}, Marks: {marks}, Grade: {grade}")
```

Output:



```
istes code/assi4ai.py"
Enter student's name: sar
Enter marks obtained: 95
Name: sar, Marks: 95, Grade: A
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter student's name: doe
Enter marks obtained: 75
Name: doe, Marks: 75, Grade: C
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter student's name: jane
Enter marks obtained: 63
Name: jane, Marks: 63, Grade: D
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter student's name: 59
Enter marks obtained: 59
Name: 59, Marks: 59, Grade: F
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter student's name: smith
Enter marks obtained: 59
Name: smith, Marks: 59, Grade: F
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter student's name: john
Enter marks obtained: 88
Name: john, Marks: 88, Grade: B
PS C:\Users\shyam\ai assistes code> 
```

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

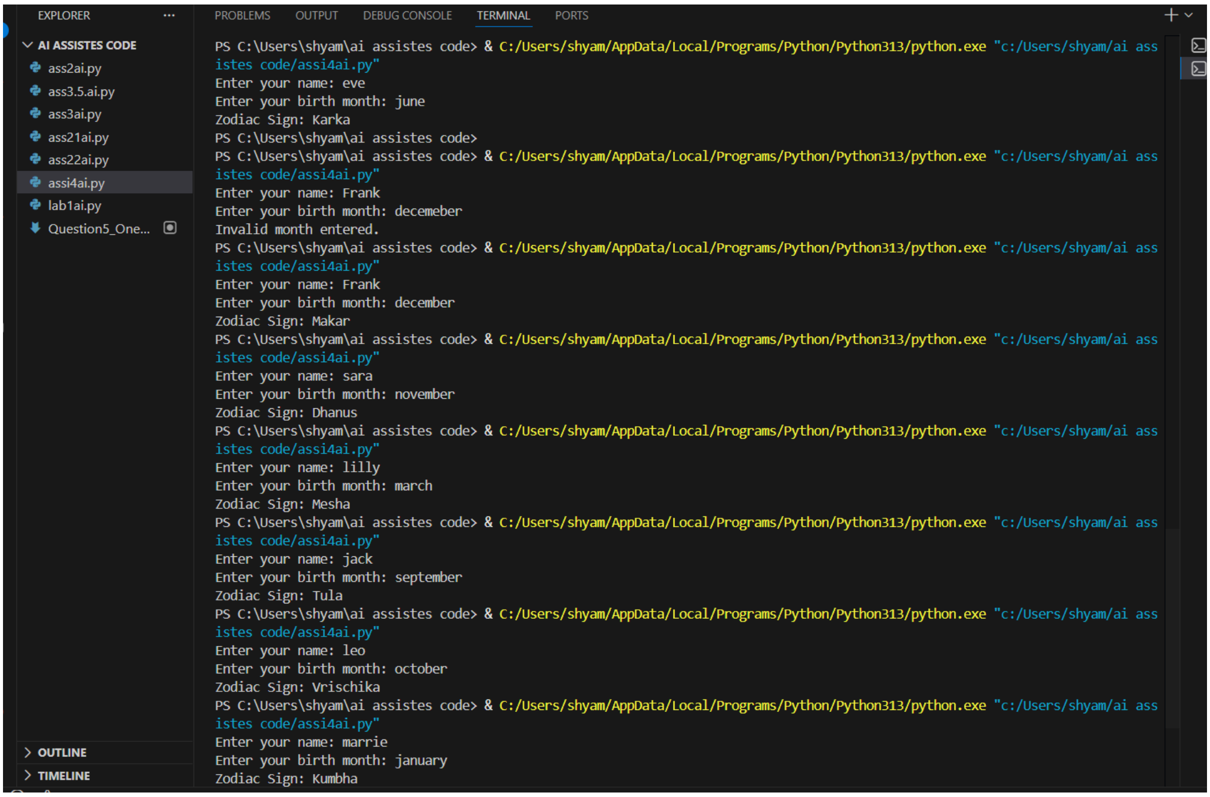
Prompt:

```
'''
input : name: "Alice", brthmonth:february output :Zodiac Sign :Meena
input : name: "Bob", brthmonth:august output :Zodiac Sign :kanya
input : name: "Charlie", brthmonth:november output :Zodiac Sign :Dhanus
input : name: "Diana", brthmonth:may output :Zodiac Sign :mithuna
input : name: "Eve", brthmonth:june output :Zodiac Sign :Karka
input : name: "Frank", brthmonth:december output :Zodiac Sign :Makar
input : name: "Grace", brthmonth:april output :Zodiac Sign :Vrishabha
input : name: "Hank", brthmonth:july output :Zodiac Sign :Simha
input : name: "Ivy", brthmonth:march output :Zodiac Sign :Mesha
input : name: "Jack", brthmonth:september output :Zodiac Sign :Tula
input : name: "Kathy", brthmonth:october output :Zodiac Sign :Vrischika
input : name: "Leo", brthmonth:january output :Zodiac Sign :Kumbha
'''
```

Code:

```python
name = input("Enter your name: ")
brthmonth = input("Enter your birth month: ").strip().lower()
zodiac_signs = {
    "january": "Kumbha",
    "february": "Meena",
    "march": "Mesha",
    "april": "Vrishabha",
    "may": "Mithuna",
    "june": "Karka",
    "july": "Simha",
    "august": "Kanya",
    "september": "Tula",
    "october": "Vrischika",
    "november": "Dhanus",
    "december": "Makar"
}
if brthmonth in zodiac_signs:
    print(f"Zodiac Sign: {zodiac_signs[brthmonth]}")
else:
    print("Invalid month entered.")
```

Output :



```
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter your name: eve
Enter your birth month: june
Zodiac Sign: Karka
PS C:\Users\shyam\ai assistes code>
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter your name: Frank
Enter your birth month: decemeber
Invalid month entered.
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter your name: Frank
Enter your birth month: december
Zodiac Sign: Makar
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter your name: sara
Enter your birth month: november
Zodiac Sign: Dhanus
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter your name: lilly
Enter your birth month: march
Zodiac Sign: Mesha
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter your name: jack
Enter your birth month: september
Zodiac Sign: Tula
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter your name: leo
Enter your birth month: october
Zodiac Sign: Vrischika
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/assi4ai.py"
Enter your name: marrie
Enter your birth month: january
Zodiac Sign: Kumbha
```

5. Result Analysis Based on Marks

Task: Write a Python program that determines whether a student

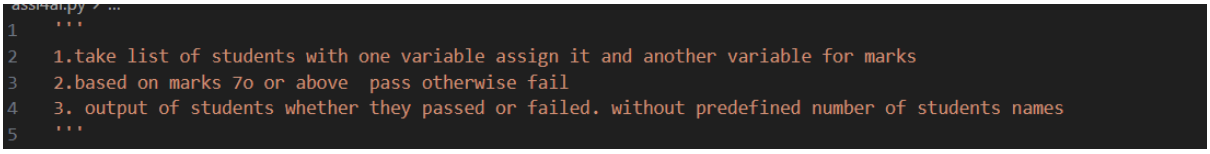
Passes or Fails based on marks using Chain-of-Thought (CoT)

prompting.

Result Categories:

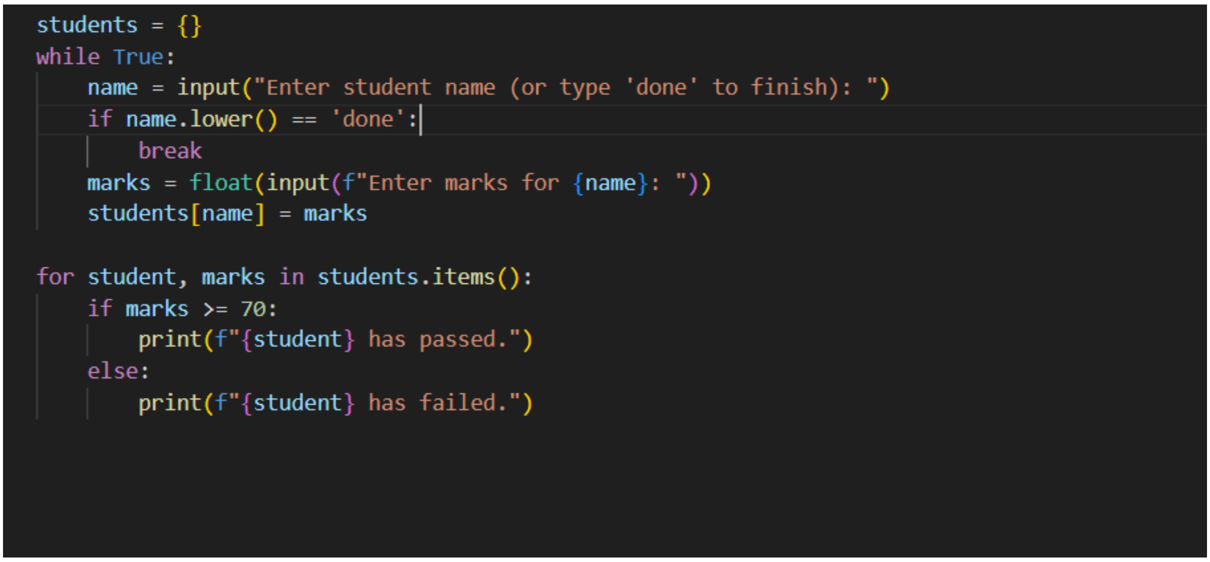
['Pass', 'Fail']

Prompt :



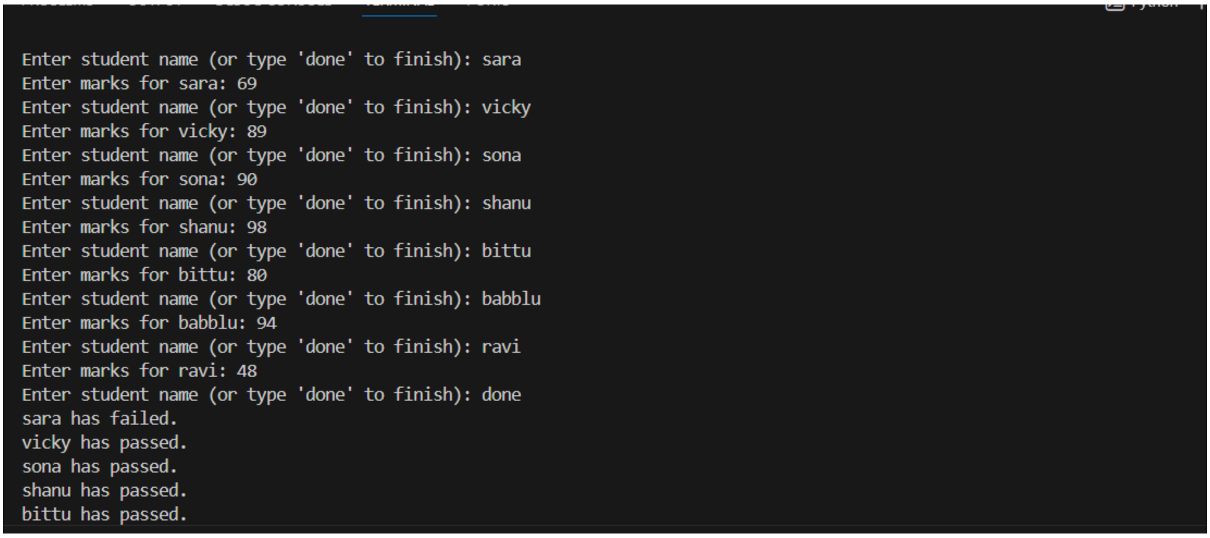
```
'''
1.take list of students with one variable assign it and another variable for marks
2.based on marks 7o or above  pass otherwise fail
3. output of students whether they passed or failed. without predefined number of students names
'''
```

Code:



```
students = {}
while True:
    name = input("Enter student name (or type 'done' to finish): ")
    if name.lower() == 'done':
        break
    marks = float(input(f"Enter marks for {name}: "))
    students[name] = marks

for student, marks in students.items():
    if marks >= 70:
        print(f"{student} has passed.")
    else:
        print(f"{student} has failed.")
```

Output :



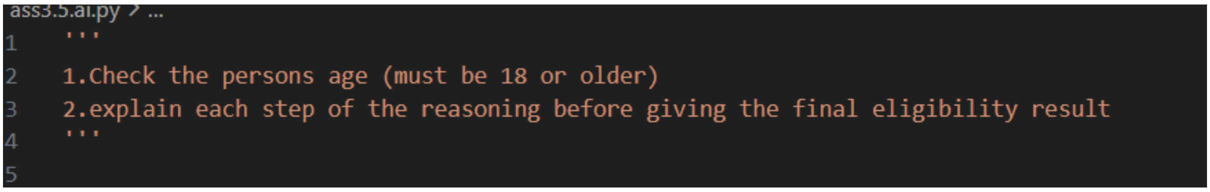
```
Enter student name (or type 'done' to finish): sara
Enter marks for sara: 69
Enter student name (or type 'done' to finish): vicky
Enter marks for vicky: 89
Enter student name (or type 'done' to finish): sona
Enter marks for sona: 90
Enter student name (or type 'done' to finish): shanu
Enter marks for shanu: 98
Enter student name (or type 'done' to finish): bittu
Enter marks for bittu: 80
Enter student name (or type 'done' to finish): babblu
Enter marks for babblu: 94
Enter student name (or type 'done' to finish): ravi
Enter marks for ravi: 48
Enter student name (or type 'done' to finish): done
sara has failed.
vicky has passed.
sona has passed.
shanu has passed.
bittu has passed.
```

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

Task: Write a Python program that determines whether a person is

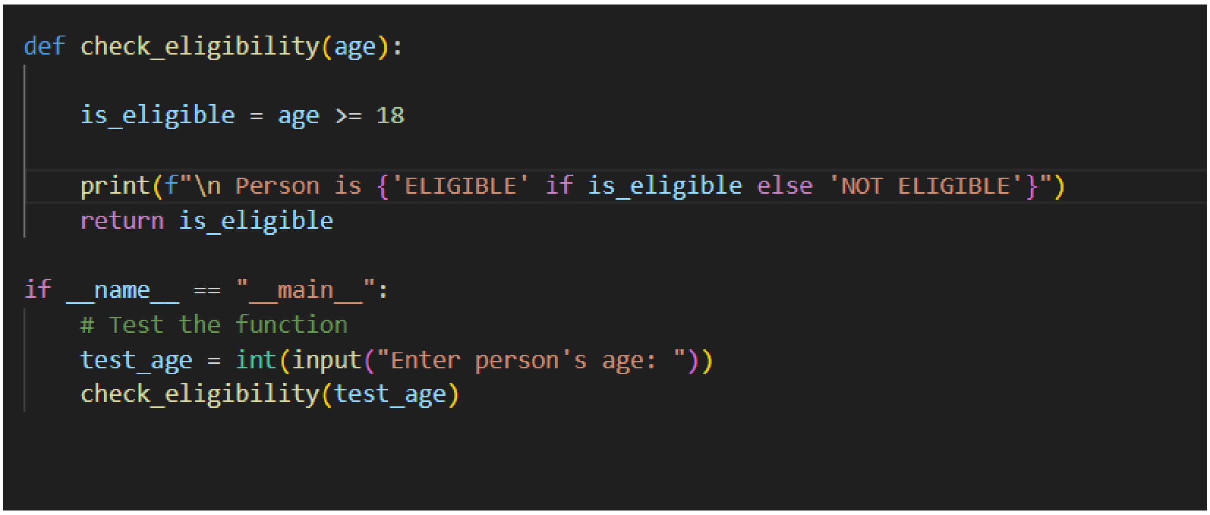
eligible to vote using Chain-of-Thought (CoT) prompting.

Prompt:



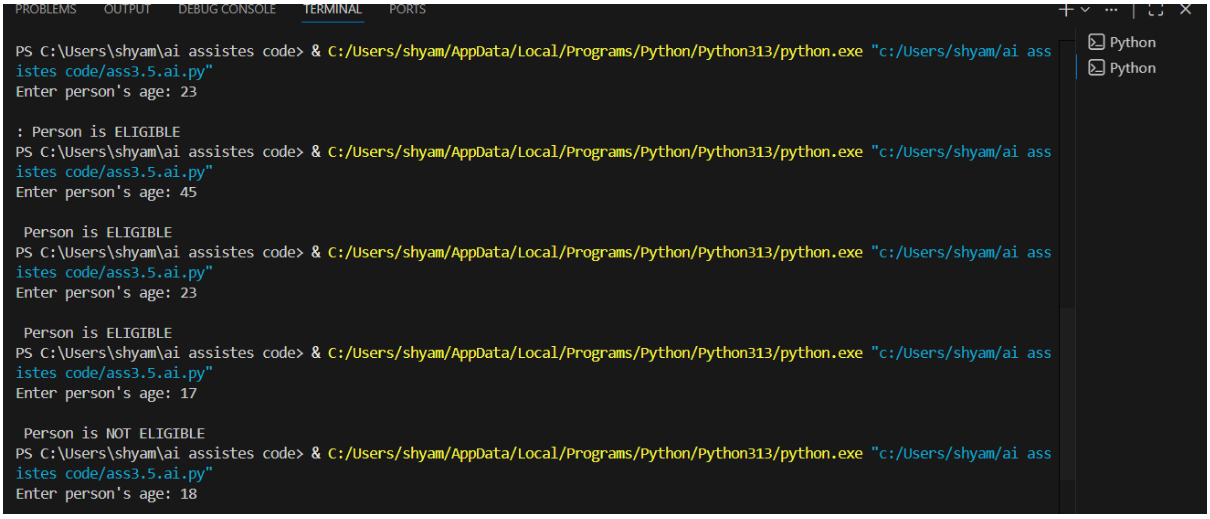
```
'''
1.Check the persons age (must be 18 or older)
2.explain each step of the reasoning before giving the final eligibility result
'''
```

Code:



```
def check_eligibility(age):

    is_eligible = age >= 18

    print(f"\n Person is {'ELIGIBLE' if is_eligible else 'NOT ELIGIBLE'}")
    return is_eligible

if __name__ == "__main__":
    # Test the function
    test_age = int(input("Enter person's age: "))
    check_eligibility(test_age)
```

Output:

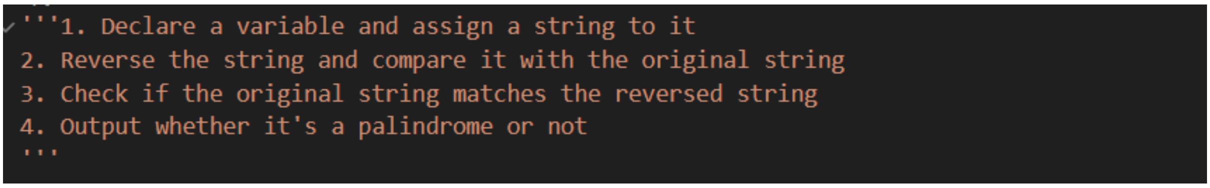


```
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/ass3.5.ai.py"
Enter person's age: 23

: Person is ELIGIBLE
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/ass3.5.ai.py"
Enter person's age: 45

 Person is ELIGIBLE
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/ass3.5.ai.py"
Enter person's age: 23

 Person is ELIGIBLE
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/ass3.5.ai.py"
Enter person's age: 17

 Person is NOT ELIGIBLE
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/ass3.5.ai.py"
Enter person's age: 18
```

Q7 Prompt Chaining (String Processing – Palindrome Names)

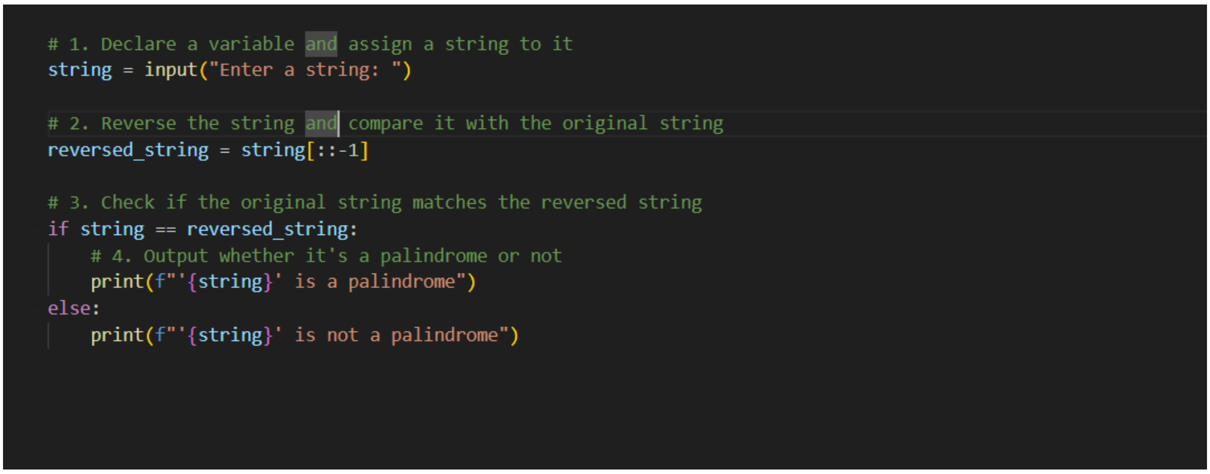
Task: Write a Python program that uses the prompt chaining technique to identify palindrome names from a list of student names.

Prompt :



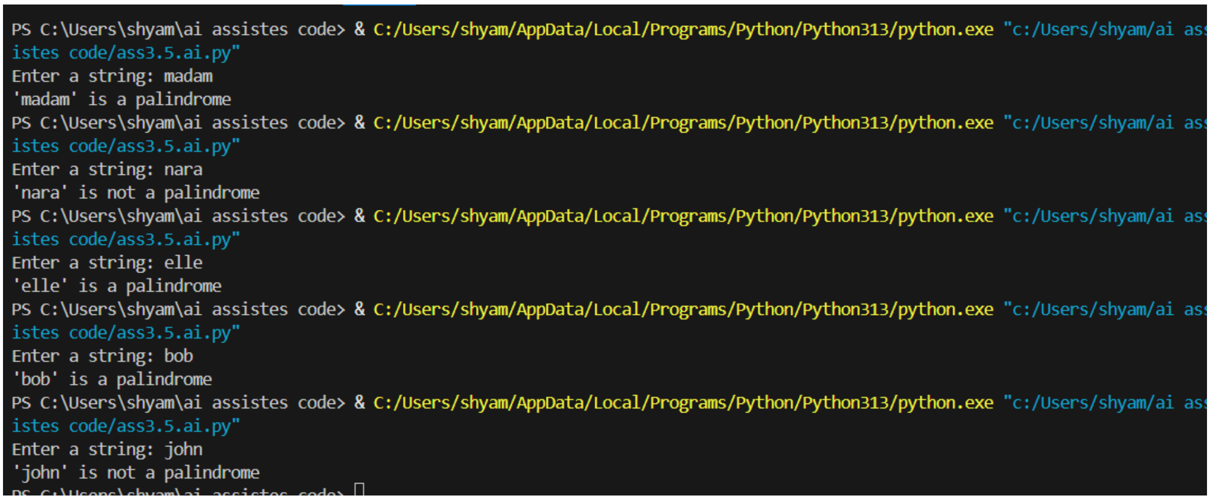
```
'''1. Declare a variable and assign a string to it
2. Reverse the string and compare it with the original string
3. Check if the original string matches the reversed string
4. Output whether it's a palindrome or not
'''
```

Code:



```python
# 1. Declare a variable and assign a string to it
string = input("Enter a string: ")

# 2. Reverse the string and compare it with the original string
reversed_string = string[::-1]

# 3. Check if the original string matches the reversed string
if string == reversed_string:
    # 4. Output whether it's a palindrome or not
    print(f"'{string}' is a palindrome")
else:
    print(f"'{string}' is not a palindrome")
```

Output :



```
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/ass3.5.ai.py"
Enter a string: madam
'madam' is a palindrome
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/ass3.5.ai.py"
Enter a string: nara
'nara' is not a palindrome
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/ass3.5.ai.py"
Enter a string: elle
'elle' is a palindrome
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/ass3.5.ai.py"
Enter a string: bob
'bob' is a palindrome
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai ass
istes code/ass3.5.ai.py"
Enter a string: john
'john' is not a palindrome
PS C:\Users\shyam\ai assistes code> 
```

8 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

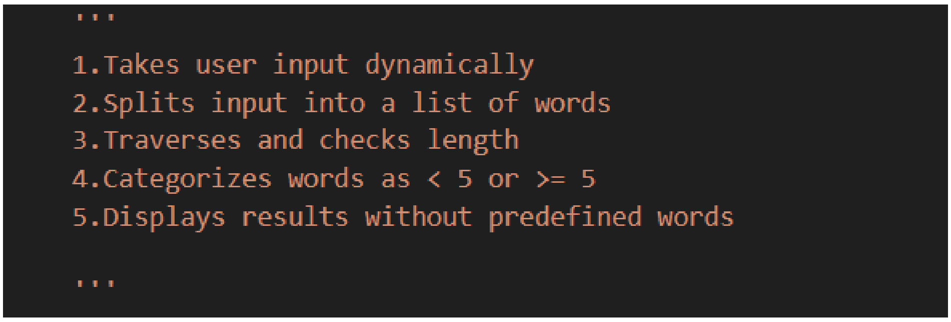
each word. In the third prompt, use the output of the previous step

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

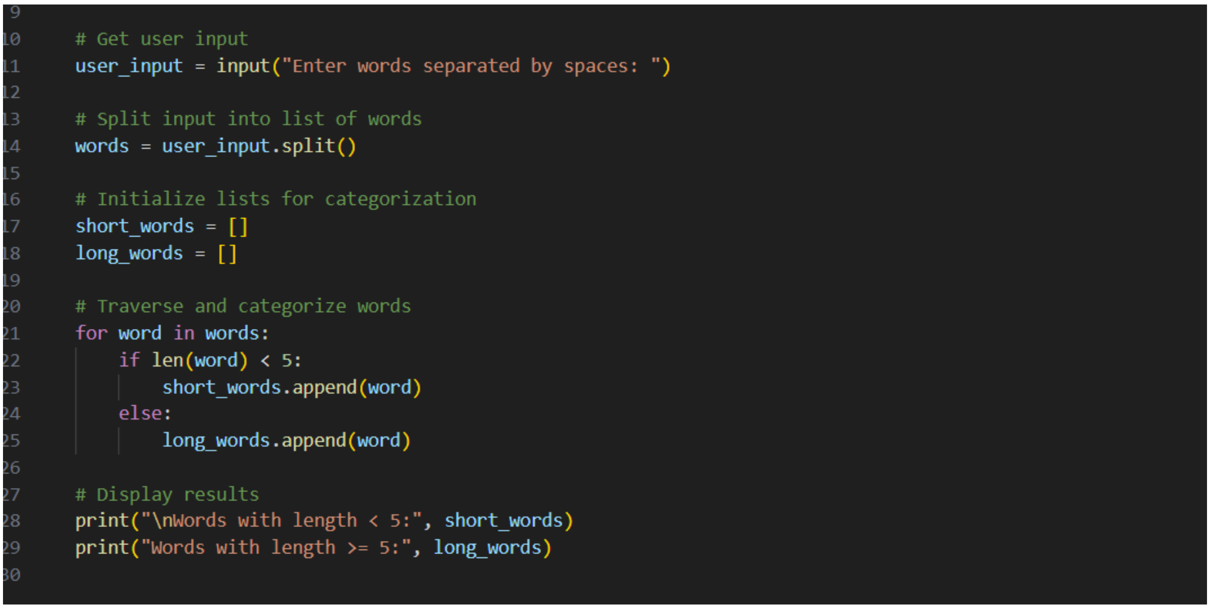
each word

Prompt :



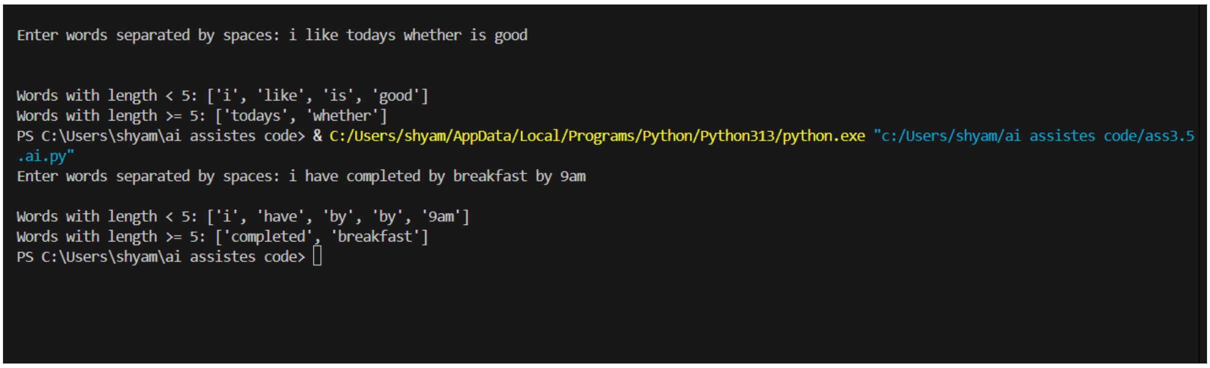
```
'''
1.Takes user input dynamically
2.Splits input into a list of words
3.Traverses and checks length
4.Categorizes words as < 5 or >= 5
5.Displays results without predefined words

'''
```

Code:



```python

# Get user input
user_input = input("Enter words separated by spaces: ")

# Split input into list of words
words = user_input.split()

# Initialize lists for categorization
short_words = []
long_words = []

# Traverse and categorize words
for word in words:
    if len(word) < 5:
        short_words.append(word)
    else:
        long_words.append(word)

# Display results
print("\nWords with length < 5:", short_words)
print("Words with length >= 5:", long_words)

```

Output:



```
Enter words separated by spaces: i like todays whether is good


Words with length < 5: ['i', 'like', 'is', 'good']
Words with length >= 5: ['todays', 'whether']
PS C:\Users\shyam\ai assistes code> & C:/Users/shyam/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/shyam/ai assistes code/ass3.5
.ai.py"
Enter words separated by spaces: i have completed by breakfast by 9am

Words with length < 5: ['i', 'have', 'by', 'by', '9am']
Words with length >= 5: ['completed', 'breakfast']
PS C:\Users\shyam\ai assistes code> 
```