# Lab Assignment-4.1

Name : G.Anoop Goud

Hallticket:2303A51085

Batch-02

**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.

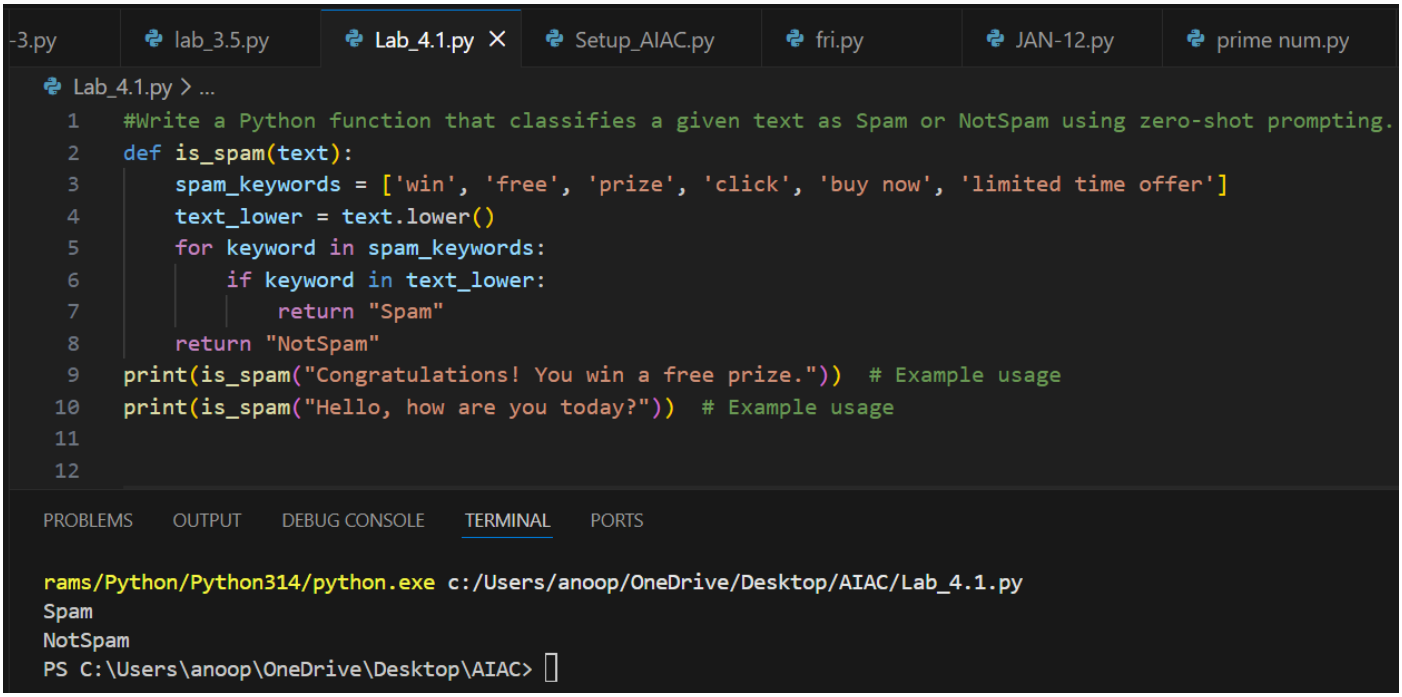
Input:
"Congratulations! You have won a free lottery ticket."

Expected Output:
Spam

Prompt:

#write a python function that classifies a given test as spam or not spam based on the presence of certain keywords

Code:



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

Prompt:

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

write a python program that detect the mood of the person and take the sentance from the user if any word from the emotions list is present in the sentance then display that emotion otherwise display no emotion detected

Code:

```python
#write a python program that detect the mood of the person and take the sentance fro
Emotions: ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']
def detect_emotion(text):
    emotions = {
        'happy': ['joy', 'pleased', 'content', 'cheerful'],
        'sad': ['unhappy', 'sorrowful', 'dejected', 'downcast'],
        'angry': ['mad', 'furious', 'irate', 'enraged'],
        'excited': ['thrilled', 'elated', 'overjoyed', 'eager'],
        'nervous': ['anxious', 'tense', 'apprehensive', 'worried'],
        'neutral': ['calm', 'unemotional', 'indifferent', 'composed']
    }
    text_lower = text.lower()
    for emotion, keywords in emotions.items():
        for keyword in keywords:
            if keyword in text_lower:
                return emotion
    return "neutral"
print(detect_emotion("I am so thrilled about the trip!"))  # Example usage
print(detect_emotion("I feel very anxious about the exam."))  # Example usage
```

```
EMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Python/Python314/python.exe c:/Users/anoop/OneDrive/Desktop/AIAC/Lab_4.1.py

am
\Users\anoop\OneDrive\Desktop\AIAC> & C:/Users/anoop/AppData/Local/Programs/Python/Python3
ed
us
\Users\anoop\OneDrive\Desktop\AIAC>
```

emotions = ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']

user_input = input("Enter a sentence to detect emotion: ").lower()

detected_emotions = [emotion for emotion in emotions if emotion in user_input]

if detected_emotions:

    print(f"Detected emotions: {', '.join(detected_emotions)}")

else:

    print("No emotion detected.")

input/output:

```
Enter a sentence to detect emotion: iam happy today
Detected emotions: happy
```

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

Prompt:

90-100=A

80-89=B

70-79=C

60-69=D

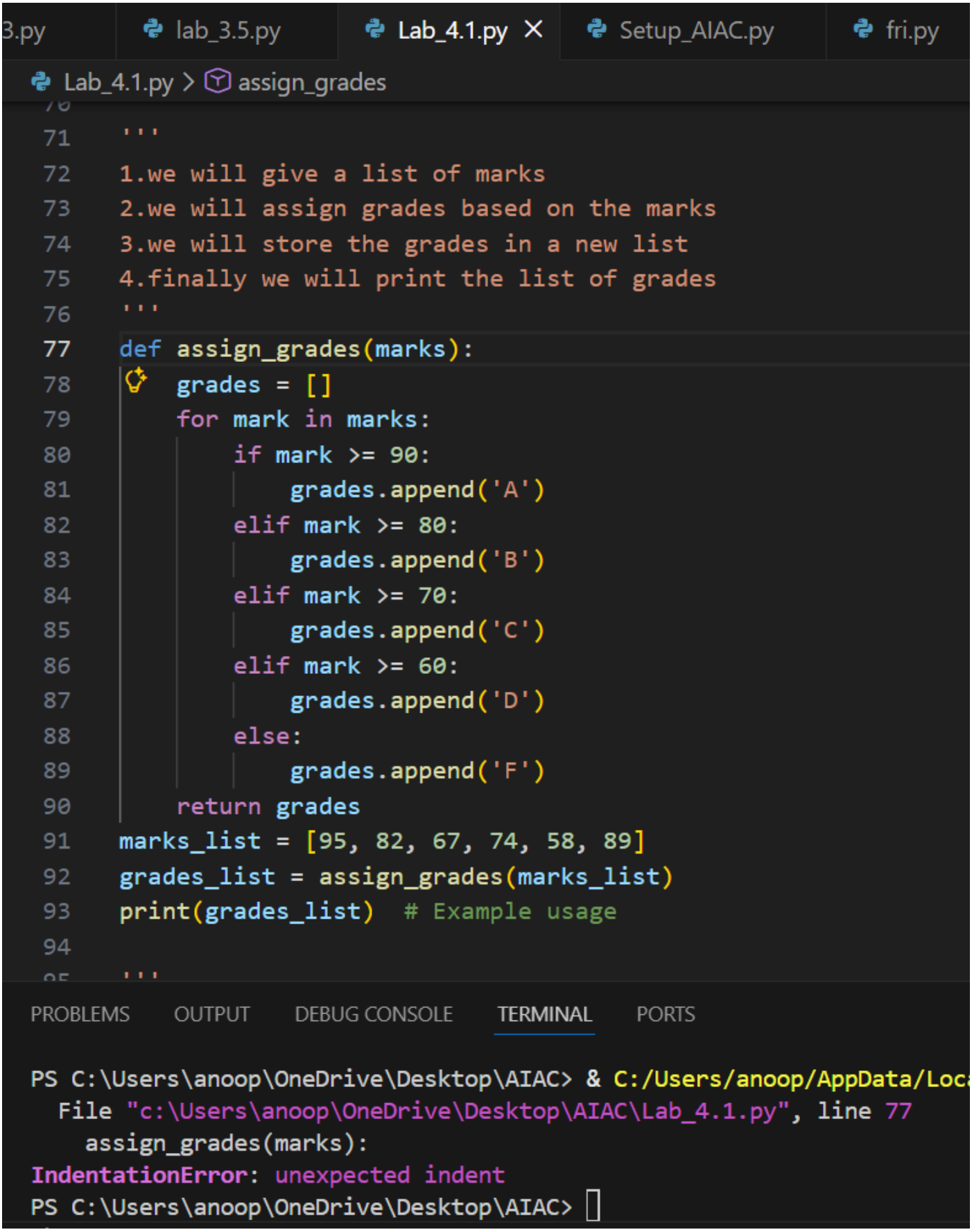
below 60=F

1.we will give a list of marks

2.we will assign grades based on the marks

3.we will store the grades in a new list

4.finally we will print the list of grades




**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:

march=mesha

april=vrishabha

may=mithuna

june=karka

july=simha

august=kanya

september=tula

october=vrischika

november=dhanus

december=makara

january=kumbha

february=meena

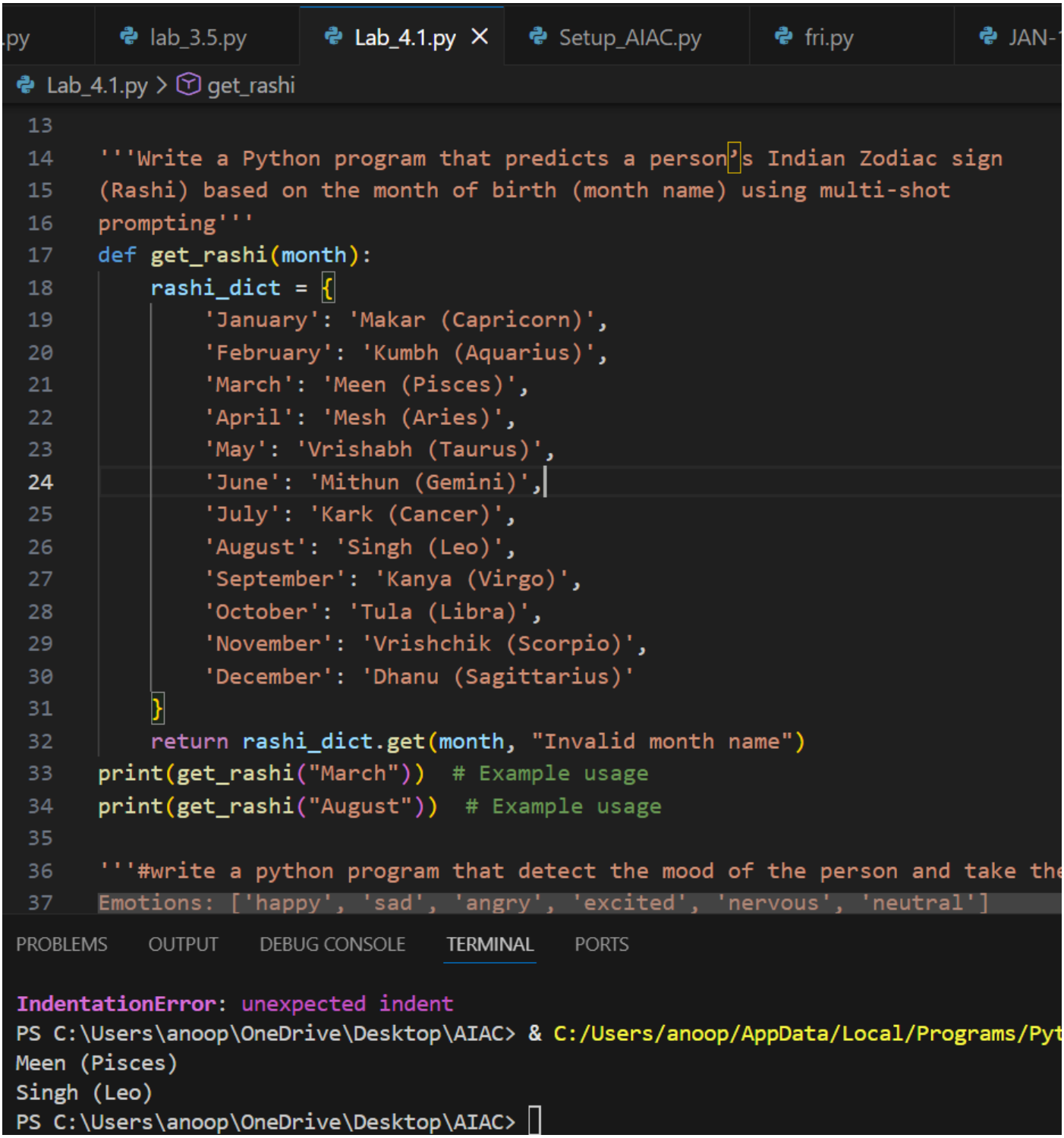
Write a Python program that predicts a person's Indian Zodiac sign

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

prompting


'''
march=mesha
april=vrishabha
may=mithuna
june=karka
july=simha
august=kanya
september=tula
october=vrischika
november=dhanus
december=makara
january=kumbha
february=meena
write a python code to accept month from the user and display t
'''
'''def get_zodiac_sign(month):

Code:




**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']

Prompt:

take marks as an input from the user

if marks is greater than or equal to 40 then print pass

if marks is less than 40 then print fail

```
"""
take marks as an input from the user
if marks is greater than or equal to 40 then print pass
if marks is less than 40 then print fail

"""
"""try:
    marks = int(input("Enter the marks: "))
    if marks < 0:
        print("Invalid input. Please enter a positive integer for marks.")
    else:
        if marks >= 40:
            print("Pass")
        else:
            print("Fail")
except ValueError:
    print("Invalid input. Please enter a valid integer for marks.")"""
```

```
"""
take marks as an input from the user
if marks is greater than or equal to 40 then print pass
if marks is less than 40 then print fail

"""
try:
    marks = int(input("Enter the marks: "))
    if marks < 0:
        print("Invalid input. Please enter a positive integer.")
    else:
        if marks >= 40:
            print("Pass")
        else:
            print("Fail")
except ValueError:
    print("Invalid input. Please enter a valid integer.")
```

Code:

```
try:
    marks = int(input("Enter the marks: "))
    if marks < 0:
        print("Invalid input. Please enter a positive integer.")
```

```
    else:
        if marks >= 40:
            print("Pass")
        else:
            print("Fail")
except ValueError:
    print("Invalid input. Please enter a valid integer.")
```

**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:

take age from the user

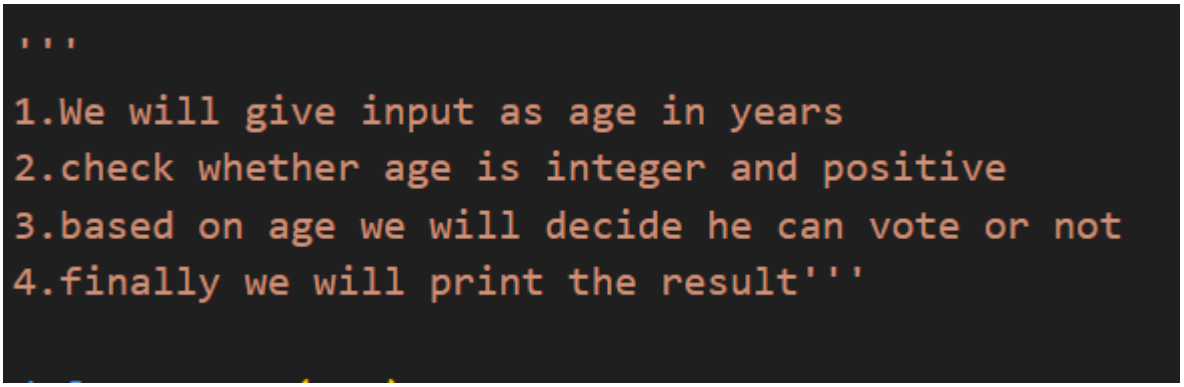
if age is equal or greater than 18

then print eligible to vote

if age is less than 18

print not eligible to vote

prompt:




Code:

```python
def can_vote(age):
    if not isinstance(age, int) or age < 0:
        return "Invalid input. Please enter a positive integer."
    if age >= 18:
        return "Eligible to vote"
    else:
        return "Not eligible to vote"
print(can_vote(20))  # Example usage
print(can_vote(16))  # Example usage
print(can_vote(-5))  # Example usage
```

try:

age = int(input("Enter your age: "))

if age < 0:

print("Invalid input. Please enter a positive integer.")

else:

if age >= 18:

print("Eligible to vote")

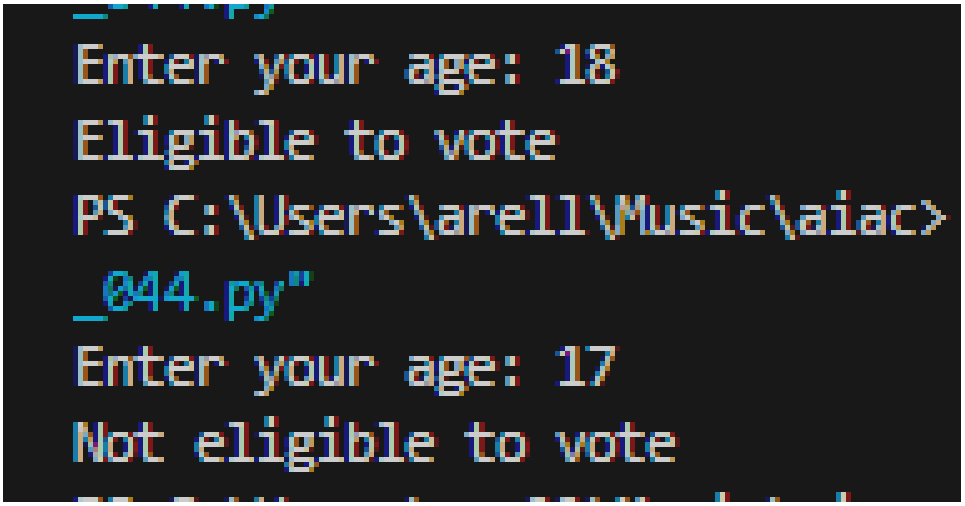
else:

print("Not eligible to vote")

except ValueError:

print("Invalid input. Please enter a valid integer.")

input/output:




**Q7 Prompt Chaining (String Processing – Palindrome Names)**

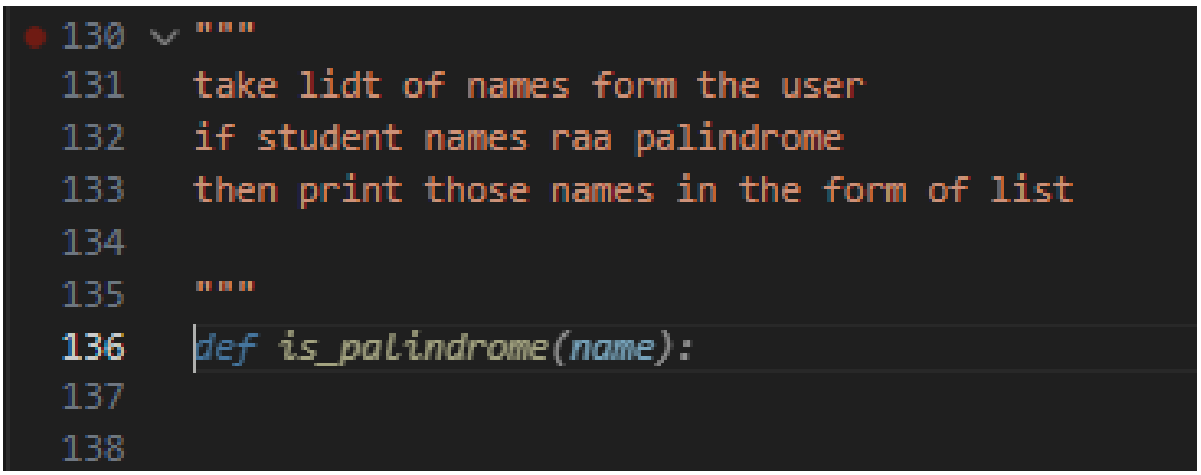
Task: Write a Python program that uses the prompt chaining technique to identify palindrome names from a list of student names.

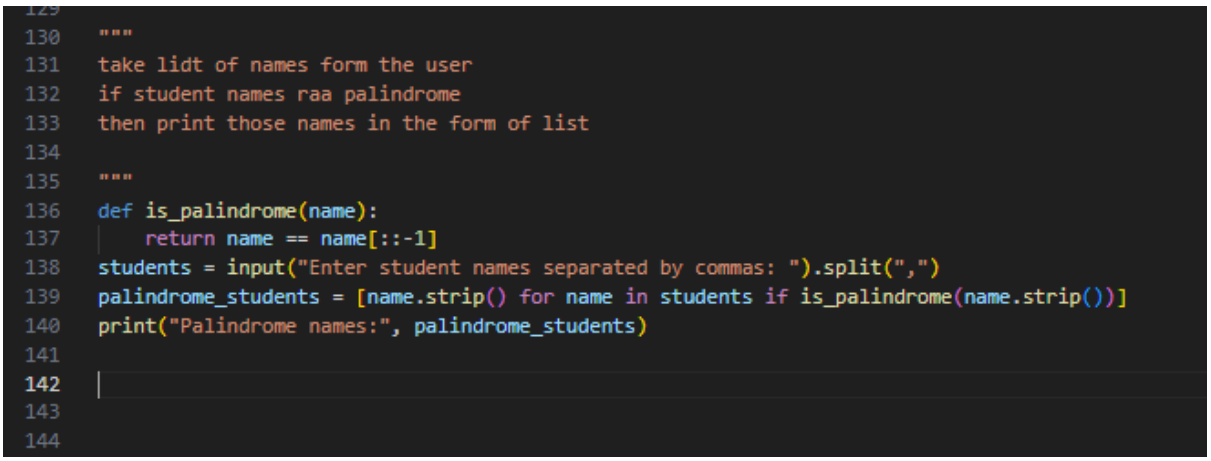
Prompt:

take list of names form the user

if student names raa palindrome

then print those names in the form of list




Code:

```
def is_palindrome(name):
    return name == name[::-1]
students = input("Enter student names separated by commas: ").split(",")
palindrome_students = [name.strip() for name in students if is_palindrome(name.strip())]
print("Palindrome names:", palindrome_students)
```

**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

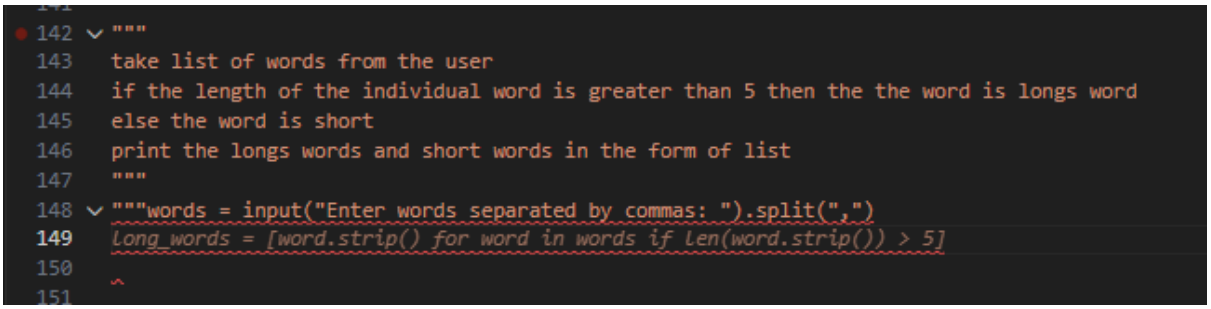
Prompt:

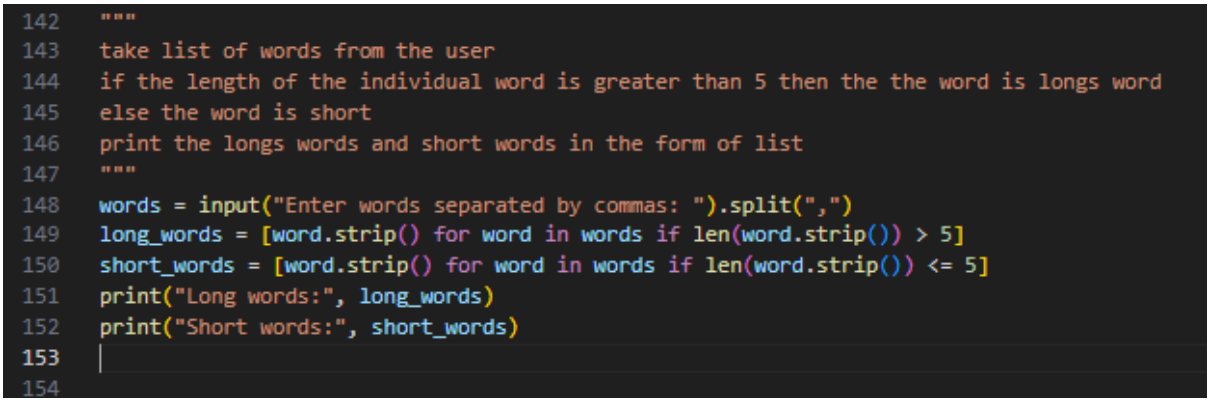
take list of words from the user

if the length of the individual word is greater than 5 then the the word is longs word

else the word is short

print the longs words and short words in the form of list




Code:

words = input("Enter words separated by commas: ").split(",")

long_words = [word.strip() for word in words if len(word.strip()) > 5]

short_words = [word.strip() for word in words if len(word.strip()) <= 5]

print("Long words:", long_words)

print("Short words:", short_words)