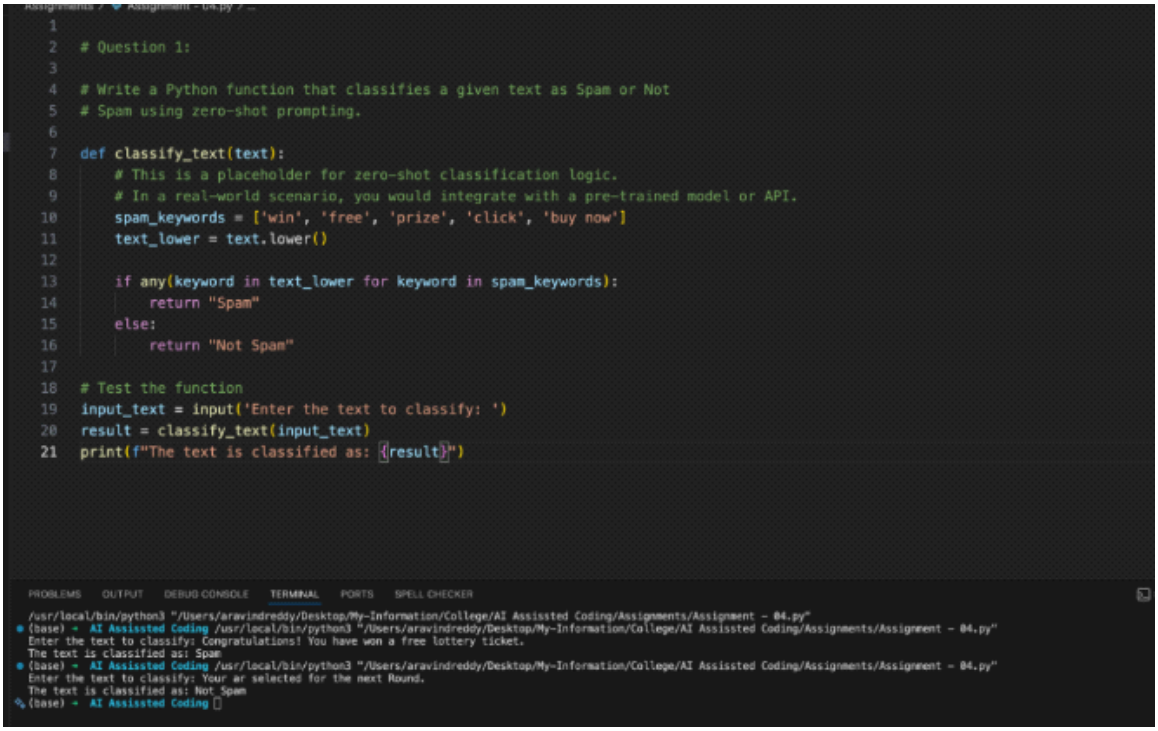
Assignment-4.1

Name:Naga Rishik Reddy

Hall ticket:2303A51089

Batch - 02

Q1. Zero-Shot Prompting (Basic Lab Task)



```python

# Question 1:

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

def classify_text(text):
    # This is a placeholder for zero-shot classification logic.
    # In a real-world scenario, you would integrate with a pre-trained model or API.
    spam_keywords = ['win', 'free', 'prize', 'click', 'buy now']
    text_lower = text.lower()

    if any(keyword in text_lower for keyword in spam_keywords):
        return "Spam"
    else:
        return "Not Spam"

# Test the function
input_text = input('Enter the text to classify: ')
result = classify_text(input_text)
print(f"The text is classified as: {result}")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SPELL CHECKER

/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Assignments/Assignment - 04.py"
(base) - AI Assissted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Assignments/Assignment - 04.py"
Enter the text to classify: Congratulations! You have won a free lottery ticket.
The text is classified as: Spam
(base) - AI Assissted Coding /usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assissted Coding/Assignments/Assignment - 04.py"
Enter the text to classify: Your ar selected for the next Round.
The text is classified as: Not Spam
(base) - AI Assissted Coding
```

Q2. One-Shot Prompting (Emotion detection)

Assignment-1.py > ...

```python
#write a python program that defects the emotion of a sentence using one-shot prompting
#emotions: ['happy', 'sad', 'angry', 'neutral']
def detect_emotion(sentence):
    #this is a placeholder for one-shot prompting logic
    #in a real-world scenario, you would integrate with a pre-trained model or API
    emotion_keywords = {
        'happy': ['joy', 'glad', 'excited', 'pleased'],
        'sad': ['unhappy', 'sorrow', 'depressed', 'down'],
        'angry': ['mad', 'furious', 'irritated', 'annoyed'],
        'neutral': []
    }

    sentence_lower = sentence.lower()
    for emotion, keywords in emotion_keywords.items():
        if any(keyword in sentence_lower for keyword in keywords):
            return emotion.capitalize()

    return "Neutral"
#test the function
input_sentence = input('Enter a sentence to detect emotion: ')
emotion_result = detect_emotion(input_sentence)
print(f"The detected emotion is: {emotion_result}")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/anaconda/python.exe "c:/Users/Rishi/Documents/Naga Rishik/AI assisted/Assignment-1.py"
Enter a sentence to detect emotion: mad
The detected emotion is: Angry
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/anaconda/python.exe "c:/Users/Rishi/Documents/Naga Rishik/AI assisted/Assignment-1.py"
Enter a sentence to detect emotion: excited
The detected emotion is: Happy
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted>
```

powershell
Python

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

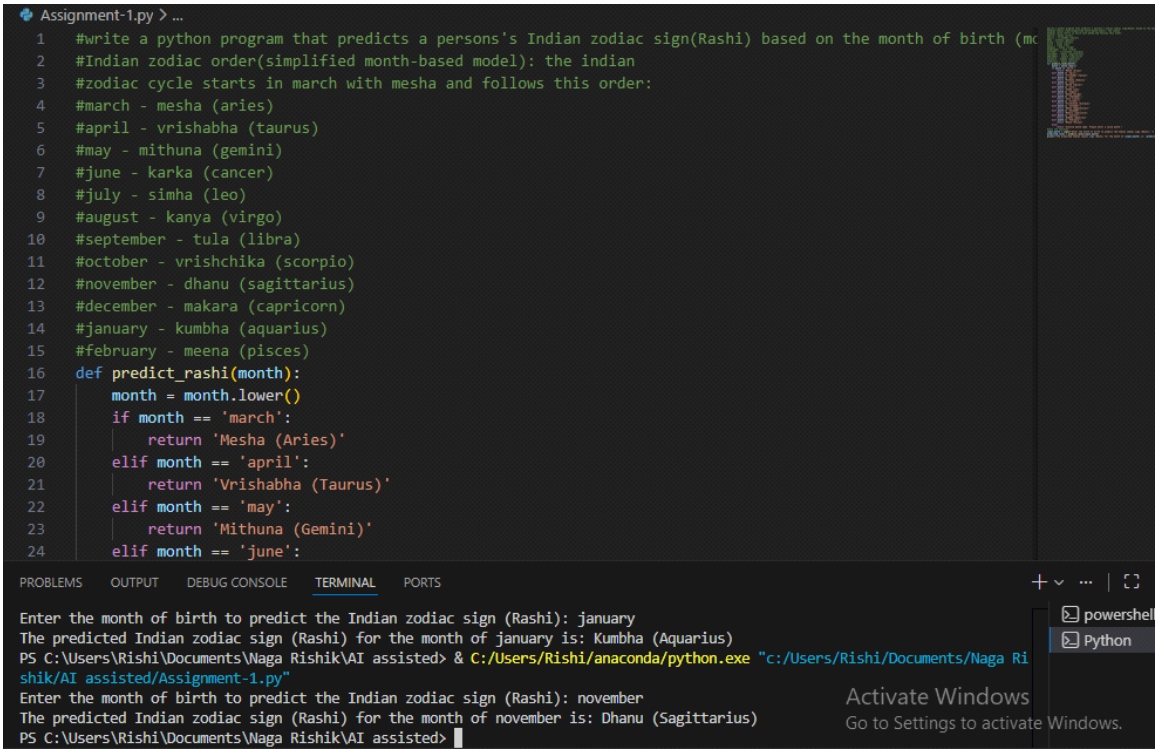
Assignment-1.py > ...

```python
#write a python program that predicts that predicts a student's 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
def predict_grade(marks):
    if marks >= 90:
        return 'A'
    elif marks >= 80:
        return 'B'
    elif marks >= 70:
        return 'C'
    elif marks >= 60:
        return 'D'
    else:
        return 'F'
#test the function
marks = float(input("Enter the student's marks: "))
grade = predict_grade(marks)
print(f"The predicted grade for the student is: {grade}")

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter the student's marks: 41
The predicted grade for the student is: F
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/anaconda/python.exe "c:/Users/Rishi/Documents/Naga Rishik/AI assisted/Assignment-1.py"
Enter the student's marks: 95
The predicted grade for the student is: A
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted>
```

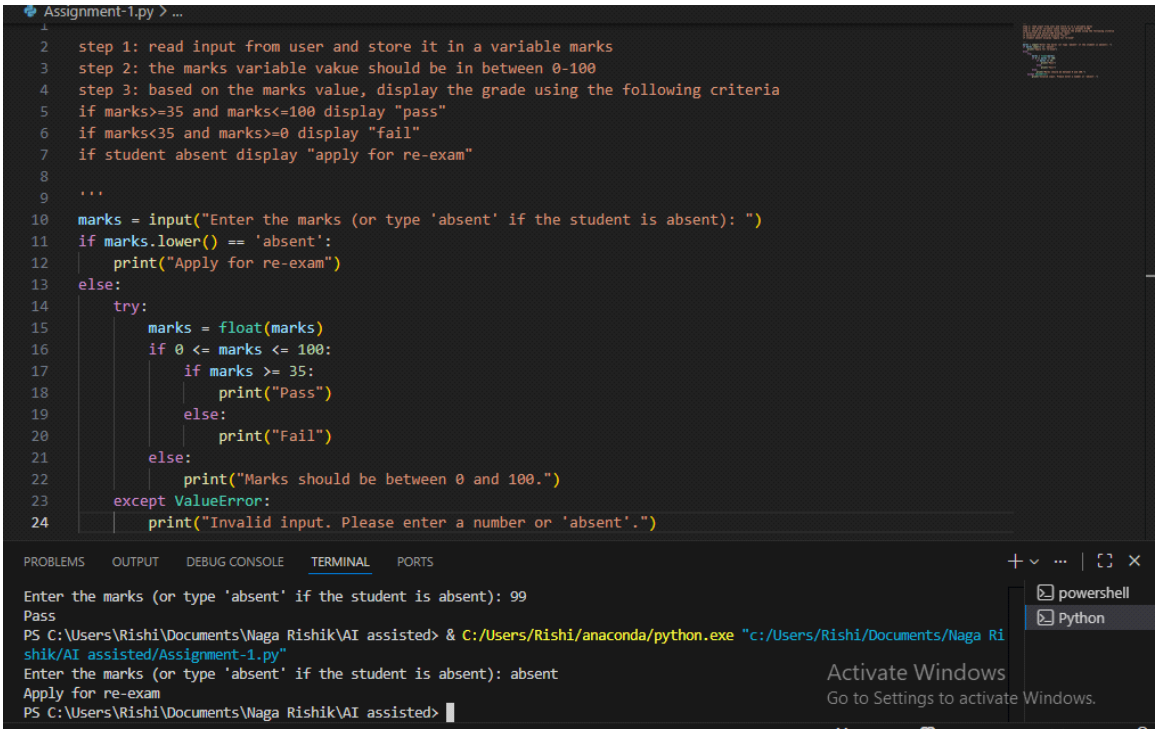
powersh
Python

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

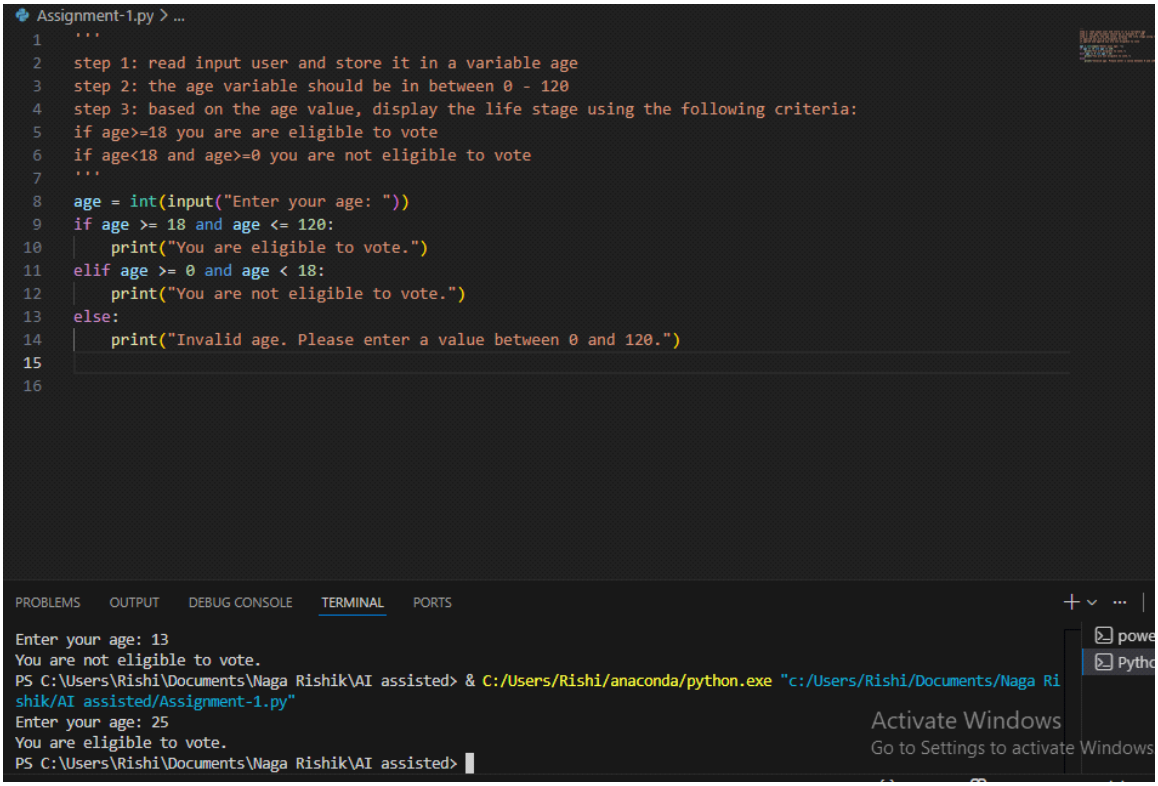


```
Assignment-1.py > ...
1   #write a python program that predicts a persons's Indian zodiac sign(Rashi) based on the month of birth (mo
2   #Indian zodiac order(simplified month-based model): the indian
3   #zodiac cycle starts in march with mesha and follows this order:
4   #march - mesha (aries)
5   #april - vrishabha (taurus)
6   #may - mithuna (gemini)
7   #june - karka (cancer)
8   #july - simha (leo)
9   #august - kanya (virgo)
10  #september - tula (libra)
11  #october - vrishchika (scorpio)
12  #november - dhanu (sagittarius)
13  #december - makara (capricorn)
14  #january - kumbha (aquarius)
15  #february - meena (pisces)
16  def predict_rashi(month):
17      month = month.lower()
18      if month == 'march':
19          return 'Mesha (Aries)'
20      elif month == 'april':
21          return 'Vrishabha (Taurus)'
22      elif month == 'may':
23          return 'Mithuna (Gemini)'
24      elif month == 'june':

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter the month of birth to predict the Indian zodiac sign (Rashi): january
The predicted Indian zodiac sign (Rashi) for the month of january is: Kumbha (Aquarius)
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted> & C:/Users/Rishi/anaconda/python.exe "c:/Users/Rishi/Documents/Naga Ri
shik/AI assisted/Assignment-1.py"
Enter the month of birth to predict the Indian zodiac sign (Rashi): november
The predicted Indian zodiac sign (Rashi) for the month of november is: Dhanu (Sagittarius)
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted>

powershell
Python

Activate Windows
Go to Settings to activate Windows.
```

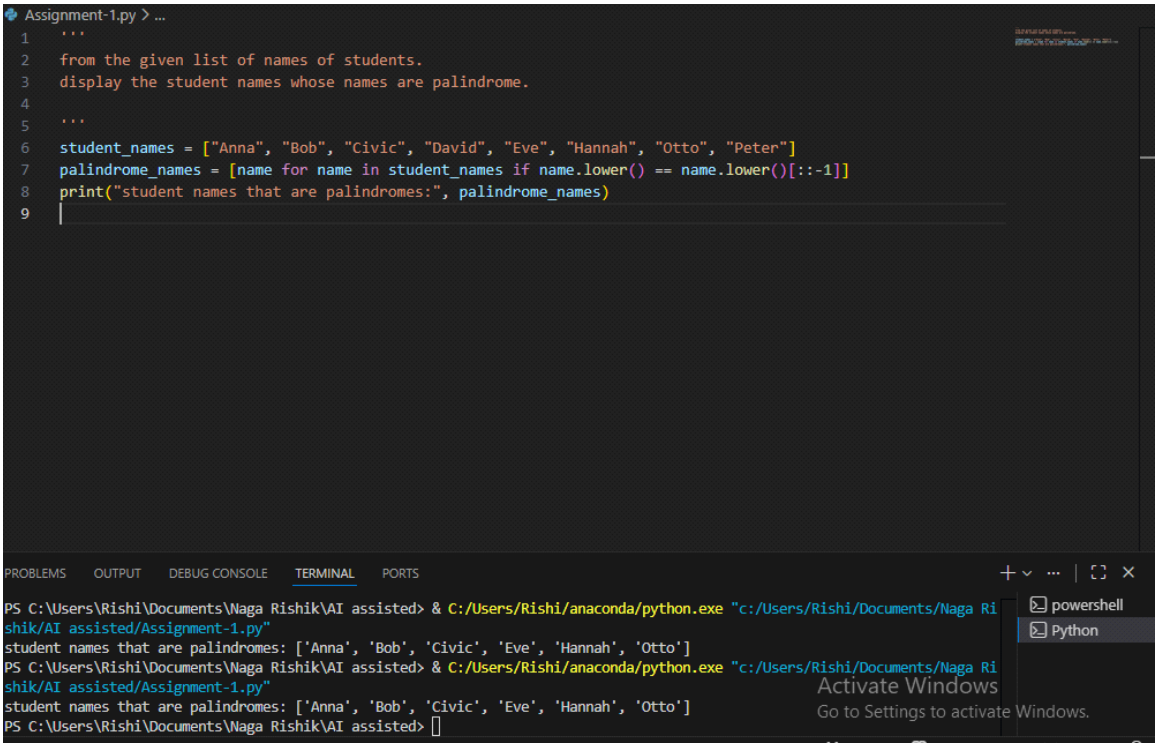
Q5. Result Analysis Based on Marks



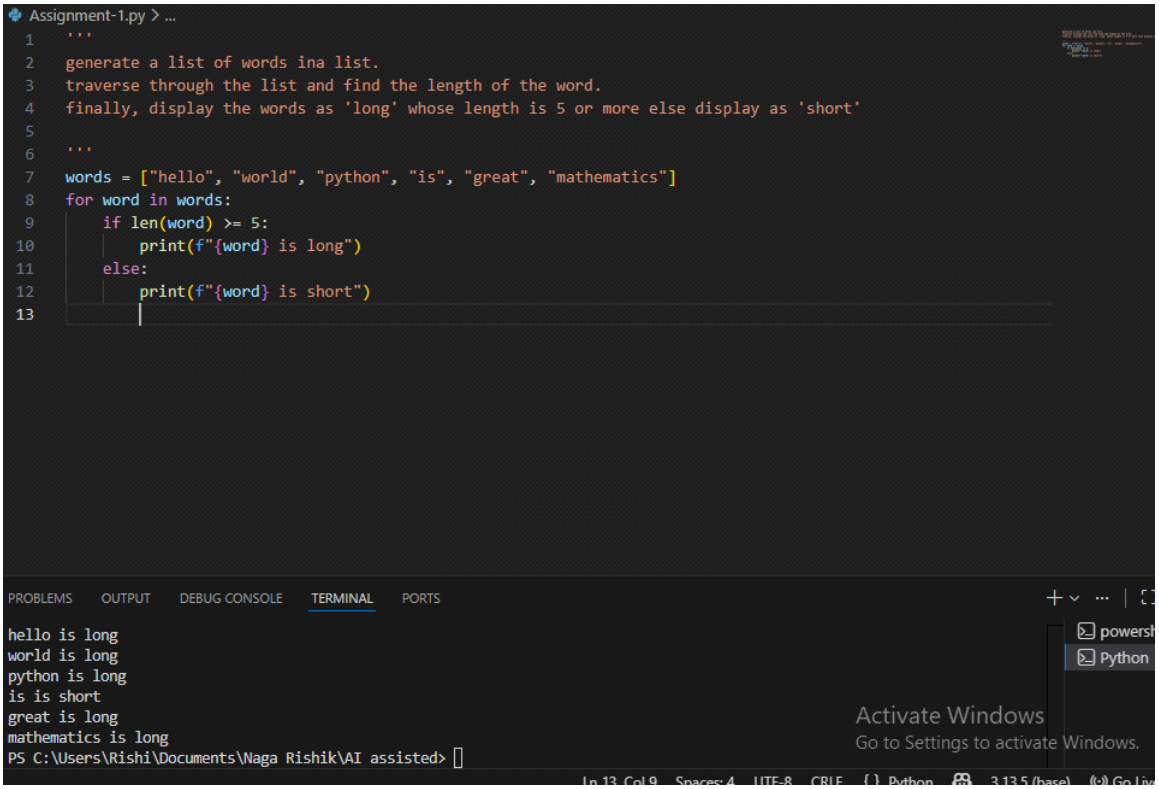
Q6 Voting Eligibility Check (Chain-of-Thought Prompting)'



Q7 Prompt Chaining (String Processing – Palindrome Names)



Q8 Prompt Chaining (String Processing – Word Length Analysis)


Assignment-1.py > ...
'''
generate a list of words ina list.
traverse through the list and find the length of the word.
finally, display the words as 'long' whose length is 5 or more else display as 'short'
'''
words = ["hello", "world", "python", "is", "great", "mathematics"]
for word in words:
if len(word) >= 5:
print(f"{word} is long")
else:
print(f"{word} is short")
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
hello is long
world is long
python is long
is is short
great is long
mathematics is long
PS C:\Users\Rishi\Documents\Naga Rishik\AI assisted>
powersh
Python
Activate Windows
Go to Settings to activate Windows.