

AI ASSISTED CODING

ASSIGNMENT-4.1

Name: G.V.N Hasini

H.T.No:2303A51099

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

```
# write a python function which reads a text and classifies the text as spam or not spam based on the presence of certain keywords like 'win',  
def classify_text_as_spam(text):  
    spam_keywords = ['win', 'prize', 'free', 'click here', 'subscribe', 'buy now']  
    text_lower = text.lower()  
  
    for keyword in spam_keywords:  
        if keyword in text_lower:  
            return "Spam"  
  
    return "Not Spam"  
# Example usage  
user_input = input("Enter the text to classify: ")  
result = classify_text_as_spam(user_input)  
print(f"The text is classified as: {result}")
```

Enter the text to classify: free
The text is classified as: Spam
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment 3"
Enter the text to classify: buy now subscribe
The text is classified as: Spam
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment 3"
Enter the text to classify: expiry date of your subscription
The text is classified as: Not Spam
PS C:\Users\hasin\OneDrive\Desktop\java programming>

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

```
"""
s=ohh! what a surprise
display excietment
user input
...
def classify_emotion(text):
    excitement_keywords = ['ohh', 'wow', 'amazing', 'fantastic', 'incredible', 'surprise']
    happiness_keywords = ['happy', 'joy', 'pleased', 'delighted', 'content']
    sadness_keywords = ['sad', 'unhappy', 'sorrow', 'dejected', 'downcast']
    angry_keywords = ['angry', 'mad', 'furious', 'irate', 'outraged']
    text_lower = text.lower()
    if any(word in text_lower for word in excitement_keywords):
        return "Excitement"
    elif any(word in text_lower for word in happiness_keywords):
        return "Happiness"
    elif any(word in text_lower for word in sadness_keywords):
        return "Sadness"
    elif any(word in text_lower for word in angry_keywords):
        return "Anger"
    else:
        return "Neutral"
user_input = input("Enter a sentence expressing an emotion: ")
emotion = classify_emotion(user_input)
print(f"The classified emotion is: {emotion}")
```

```
Enter a sentence expressing an emotion: i am so happy
The classified emotion is: Happiness
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment 3"
Enter a sentence expressing an emotion: am annoyed
The classified emotion is: Neutral
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment 3"
Enter a sentence expressing an emotion: so annoyed
The classified emotion is: Neutral
PS C:\Users\hasin\OneDrive\Desktop\java programming>
```

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

```
m=85
display B
m=55
display F
m=78
display C
m=61
display D
user input
'''

def classify_grade(m):
    if m >= 80:
        return 'A'
    elif m >= 70:
        return 'B'
    elif m >= 60:
        return 'C'
    elif m >= 50:
        return 'D'
    else:
        return 'F'

# Test cases
print(classify_grade(85)) # Expected output: B
print(classify_grade(55)) # Expected output: F
print(classify_grade(78)) # Expected output: C
print(classify_grade(61)) # Expected output: D
m = int(input("Enter the marks: "))
print(classify_grade(m))
```

```
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment_3"
A
D
B
C
Enter the marks: 91
A
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment_3"
A
D
B
C
Enter the marks: 55
A
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment_3"
A
D
B
C
Enter the marks: 89
A
```

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

```
'''  
March  
display Mesha  
April  
display Vrishabha  
May  
display Mithuna  
June  
display Karka  
July  
display Simha  
August  
display Kanya  
September  
display Tula  
October  
display Vrischika  
November  
display Dhanu  
December  
display Makara  
January  
display Kumbha  
February  
display Meena  
user inputs month name and program displays corresponding zodiac sign.
```

```

def zodiac_sign(month):
    month = month.lower()
    if month == "march":
        return "Mesha"
    elif month == "april":
        return "Vrishabha"
    elif month == "may":
        return "Mithuna"
    elif month == "june":
        return "Karka"
    elif month == "july":
        return "Simha"
    elif month == "august":
        return "Kanya"
    elif month == "september":
        return "Tula"
    elif month == "october":
        return "Vrischika"
    elif month == "november":
        return "Dhanu"
    elif month == "december":
        return "Makara"
    elif month == "january":
        return "Kumbha"
    elif month == "february":
        return "Meena"
    else:
        return "Invalid month name"
user_month = input("Enter the month name: ")
sign = zodiac_sign(user_month)
print(f"The zodiac sign for {user_month} is: {sign}")

```

```

Enter the month name: february
The zodiac sign for february is: Meena
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment 3"
Enter the month name: october
The zodiac sign for october is: Vrischika
PS C:\Users\hasin\OneDrive\Desktop\java programming>

```

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']

```
Significant 3 > ...
...
Read Student Marks
validate that the marks are between 0 and 100
display appropriate message if invalid
if marks are greater than or equal to 40, display "Pass"
else display "Fail"
user input
...
def classify_marks():
    try:
        marks = float(input("Enter the student's marks (0-100): "))
        if marks < 0 or marks > 100:
            print("Invalid marks. Please enter a value between 0 and 100.")
        elif marks >= 40:
            print("Pass")
        else:
            print("Fail")
    except ValueError:
        print("Invalid input. Please enter a numeric value.")
```

```
-1 -> if: --!-`'
Enter the student's marks (0-100): 50
Pass
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment 3"
Enter the student's marks (0-100): -1
Invalid marks. Please enter a value between 0 and 100.
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment 3"
Enter the student's marks (0-100): 111
Invalid marks. Please enter a value between 0 and 100.
PS C:\Users\hasin\OneDrive\Desktop\java programming>
```

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.

```
'''  
read the age of a person  
display eligible to vote or not based on the age being greater than or equal to 18  
invalid age if age is negative  
user input  
'''  
  
def check_voting_eligibility():  
    age = int(input("Enter your age: "))  
    if age < 0:  
        print("Invalid age")  
    elif age >= 18:  
        print("Eligible to vote")  
    else:  
        print("Not eligible to vote")  
check_voting_eligibility()  
  
Enter your age: 18  
Eligible to vote  
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment 3"  
Enter your age: 16  
Not eligible to vote  
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assignment 3"  
Enter your age: -8  
Invalid age  
PS C:\Users\hasin\OneDrive\Desktop\java programming>
```

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.

```
'''  
generate a list of names and store it in a variable called names.  
traverse through each name and make a new list of name reversing the character in each name.  
compare the original list and the new list and print the names which are same in both lists.  
user input  
'''  
  
names = ["anna", "bob", "carol", "dave", "eve", "hannah"]  
reversed_names = [name[::-1] for name in names]  
same_names = [name for name in names if name in reversed_names]  
print("Names that are the same when reversed:", same_names)  
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assig  
gment_3"  
Names that are the same when reversed: ['anna', 'bob', 'eve', 'hannah']  
PS C:\Users\hasin\OneDrive\Desktop\java programming>
```

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

```
'''  
generate a list of names  
traverse through each name and calculate the length of each name and store in variable le_word  
if le_word is 5 display short name  
if the le_word is more than or equal to 5 display long name  
'''  
  
names = ["Alice", "Bob", "Catherine", "David", "Eleanor"]  
for name in names:  
    le_word = len(name)  
    if le_word < 5:  
        print(f"{name} is a short name.")  
    else:  
        print(f"{name} is a long name.")  
PS C:\Users\hasin\OneDrive\Desktop\java programming> & C:\Users\hasin\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/hasin/OneDrive/Desktop/java programming/assig  
gment_3"  
Alice is a long name.  
Bob is a short name.  
Catherine is a long name.  
David is a long name.  
Eleanor is a long name.  
PS C:\Users\hasin\OneDrive\Desktop\java programming>
```