ASSIGNMENT -4 2303A51687

QUESTIONS _ 1-8:

1.ZERO SHOT:

2.ONE SHOT:

3.FEW SHOT:

4. MULTIPLE SHOT:

5. Result Analysis Based on Marks

6.V oting Eligibility Check (Chain-of-Thought Prompting

7.PROMPT CHAINING

```
Classify the following text as either 'Spam' or 'Not Spam'.
Text: "Congratulations! You have won a free lottery ticket."
Answer:
"""
def classify_text_prompt(text):
    spam_keywords = ['win', 'free', 'prize', 'click', 'buy now', 'limited time']
    text_lower = text.lower()
    for keyword in spam_keywords:
        if keyword in text_lower:
            return 'Spam'
    return 'Not Spam'
text = "Congratulations! You have won a free lottery ticket."
print(classify_text_prompt(text))  # Output: Spam


#Classify the following text as either 'Spam' or 'Not Spam' using zero-shot prompting.
#Detect the emotion of the given sentence.
def detect_emotion(sentence):
    emotions = {
        'happy': ['joy', 'pleased', 'content', 'cheerful'],
        'sad': ['unhappy', 'sorrowful', 'dejected', 'downcast'],
        'angry': ['mad', 'furious', 'irate', 'enraged'],
        'fearful': ['afraid', 'scared', 'anxious', 'nervous']
    }

    sentence_lower = sentence.lower()
    for emotion, keywords in emotions.items():
        for keyword in keywords:
            if keyword in sentence_lower:
                return emotion.capitalize()
    return 'Neutral'

# Predict the Indian Zodiac sign (Rashi) based on the month of birth.
def predict_rashi(month):
    rashi_dict = {
        1: 'Makar (Capricorn)',
        2: 'Kumbh (Aquarius)',
        3: 'Meen (Pisces)',
        4: 'Mesh (Aries)',
        5: 'Vrishabh (Taurus)',
        6: 'Mithun (Gemini)',
        7: 'Kark (Cancer)',
        8: 'Singh (Leo)',
        9: 'Kanya (Virgo)',
        10: 'Tula (Libra)',
        11: 'Vrishchik (Scorpio)',
        12: 'Dhanu (Sagittarius)'
    }
    return rashi_dict.get(month, 'Invalid month')
```

```python
#Predict the grade of a student based on marks. Grades: ['A', 'B', 'C', 'D', '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'
student_marks = 85
grade = predict_grade(student_marks)
print(f"The grade for marks {student_marks} is: {grade}")

#Determine whether the student passes or fails based on marks.
#Reason step by step before giving the final answer.
def pass_fail(marks):
    if marks >= 40:
        return 'Pass'
    else:
        return 'Fail'
student_marks = 35
result = pass_fail(student_marks)
print(f"The student has: {result}")
def classify_text_zero_shot(text):
    spam_keywords = ['win', 'free', 'prize', 'click', 'buy now', 'limited time']
    text_lower = text.lower()
    for keyword in spam_keywords:
        if keyword in text_lower:
            return 'Spam'
    return 'Not Spam'
text1 = "Congratulations! You win a free prize."
text2 = "Hello, how are you today?"
print(classify_text_zero_shot(text1))  # Output: Spam
print(classify_text_zero_shot(text2))  # Output: Not Spam

#Determine if a person is eligible to vote in India.
#Eligibility rule: Age must be 18 or above.
#Reason step by step before giving the final answer
def is_eligible_to_vote(age):
    if age >= 18:
        return 'Eligible to vote'
    else:
        return 'Not eligible to vote'
person_age = 20
eligibility = is_eligible_to_vote(person_age)
print(f"The person is: {eligibility}")
```

```
    if age >= 18:
        return 'Eligible to vote'
    else:
        return 'Not eligible to vote'
person_age = 20
eligibility = is_eligible_to_vote(person_age)
print(f"The person is: {eligibility}")

names = ["radar", "level", "arun", "madam", "sita"]
#Identify palindrome names from the list.
Names: ['radar', 'level', 'arun', 'madam', 'sita']
def identify_palindromes(names):
    palindromes = []
    for name in names:
        if name.lower() == name.lower()[::-1]:
            palindromes.append(name)
    return palindromes
palindrome_names = identify_palindromes(names)
print(f"Palindrome names: {palindrome_names}")


prompt = """
Calculate the length of each word.
Words: ['apple', 'dog', 'banana', 'cat', 'elephant']
Answer:
apple → 5
dog → 3
banana → 6
cat → 3
elephant → 8
"""
def calculate_word_lengths(words):
    lengths = {}
    for word in words:
        lengths[word] = len(word)
    return lengths
words = ['apple', 'dog', 'banana', 'cat', 'elephant']
word_lengths = calculate_word_lengths(words)
print(word_lengths)
```

```
PS C:\Users\adars> & C:/Users/adars/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/adars/OneDrive/New folder/HTML/AI & AC.py"
Spam
The Rashi for month 5 is: Vrishabh (Taurus)
The grade for marks 85 is: B
The student has: Fail
Spam
Not Spam
The person is: Eligible to vote
Palindrome names: ['radar', 'level', 'madam']
{'apple': 5, 'dog': 3, 'banana': 6, 'cat': 3, 'elephant': 8}
PS C:\Users\adars>
```