# Lab Assignment-4.1

**Name:** M. Hasini

**Ht.no:** 2303A51109

**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:

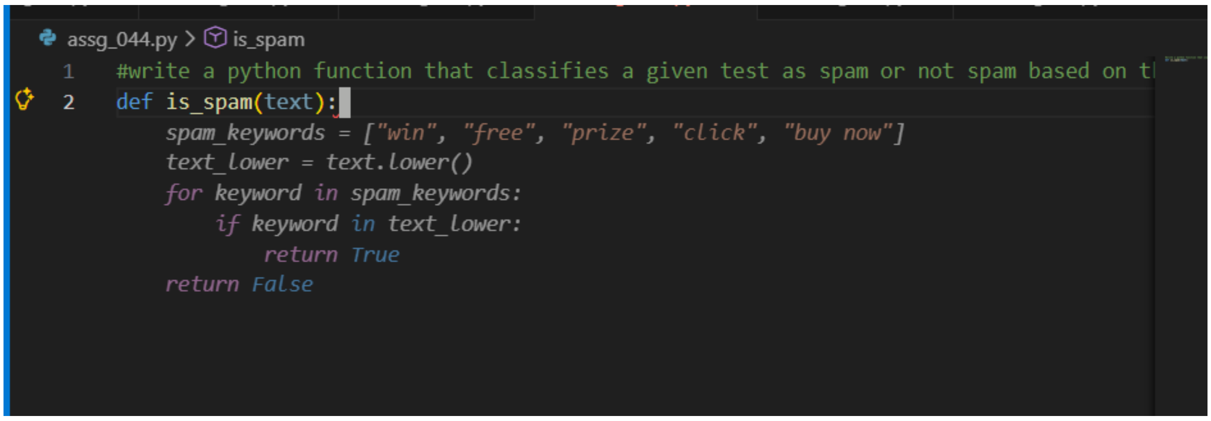
1. Construct a prompt without any examples.
2. Clearly specify the output labels.
3. Display only the predicted label.

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

Expected Output:
Spam

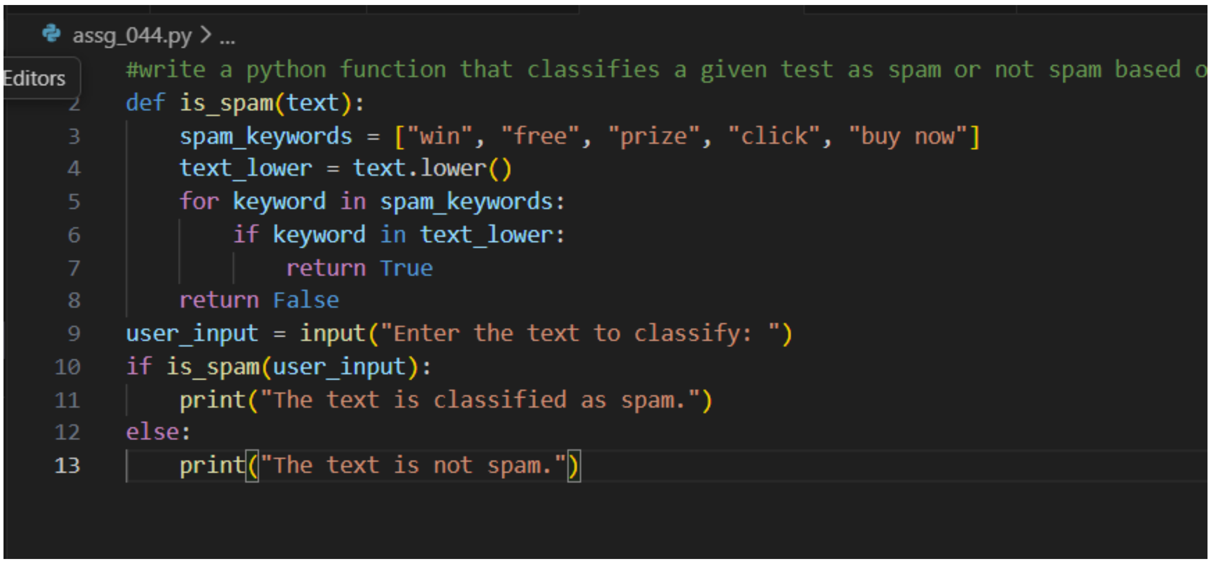
Prompt:

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



assg_044.py > is_spam

```python
#write a python function that classifies a given test as spam or not spam based on t
def is_spam(text):
    spam_keywords = ["win", "free", "prize", "click", "buy now"]
    text_lower = text.lower()
    for keyword in spam_keywords:
        if keyword in text_lower:
            return True
    return False
```

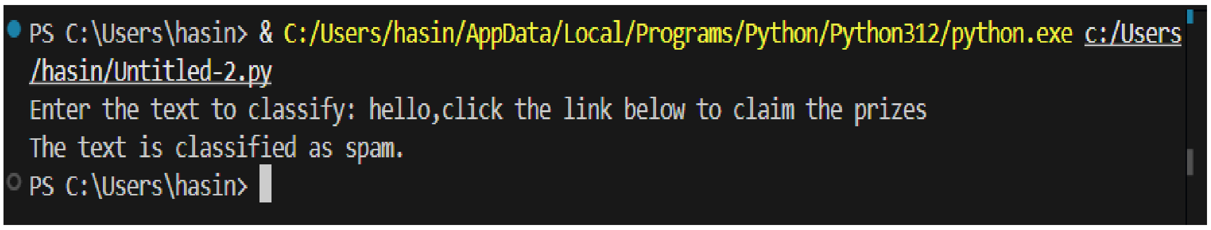


```
assg_044.py > ...
Editors
#write a python function that classifies a given test as spam or not spam based o
def is_spam(text):
    spam_keywords = ["win", "free", "prize", "click", "buy now"]
    text_lower = text.lower()
    for keyword in spam_keywords:
        if keyword in text_lower:
            return True
    return False
user_input = input("Enter the text to classify: ")
if is_spam(user_input):
    print("The text is classified as spam.")
else:
    print("The text is not spam.")
```

**Code:**

```
def is_spam(text):
    spam_keywords = ["win", "free", "prize", "click", "buy now"]
    text_lower = text.lower()
    for keyword in spam_keywords:
        if keyword in text_lower:
            return True
    return False
user_input = input("Enter the text to classify: ")
if is_spam(user_input):
    print("The text is classified as spam.")
else:
    print("The text is not spam.")
```

input/output:



```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter the text to classify: hello,click the link below to claim the prizes
The text is classified as spam.
PS C:\Users\hasin>
```

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’]

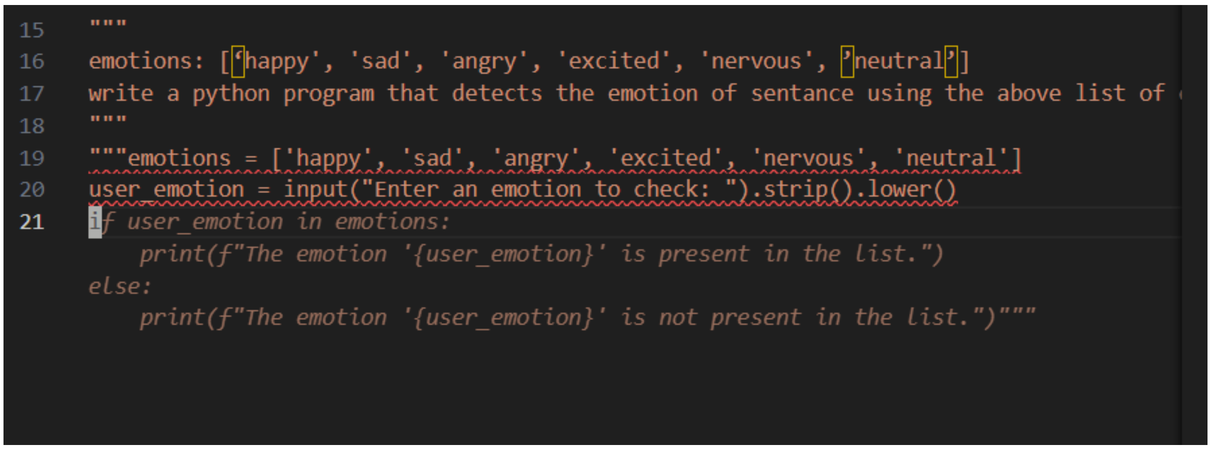
Steps:

1. Provide one labeled example inside the prompt.
2. Take a sentence as input.
3. Print the predicted emotion

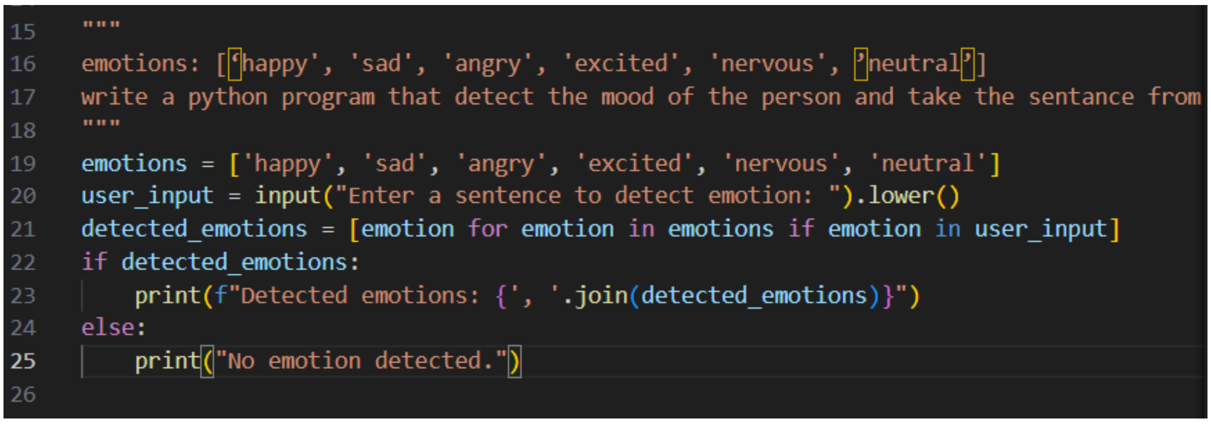
Prompt:

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.



```
"""
emotions: [‘happy', 'sad', 'angry', 'excited', 'nervous', ’neutral’]
write a python program that detects the emotion of sentence using the above list of
"""
"""emotions = ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']
user_emotion = input("Enter an emotion to check: ").strip().lower()
if user_emotion in emotions:
    print(f"The emotion '{user_emotion}' is present in the list.")
else:
    print(f"The emotion '{user_emotion}' is not present in the list.")"""
```



```
"""
emotions: [‘happy', 'sad', 'angry', 'excited', 'nervous', ’neutral’]
write a python program that detect the mood of the person and take the sentance from
"""
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.")
```

Code:

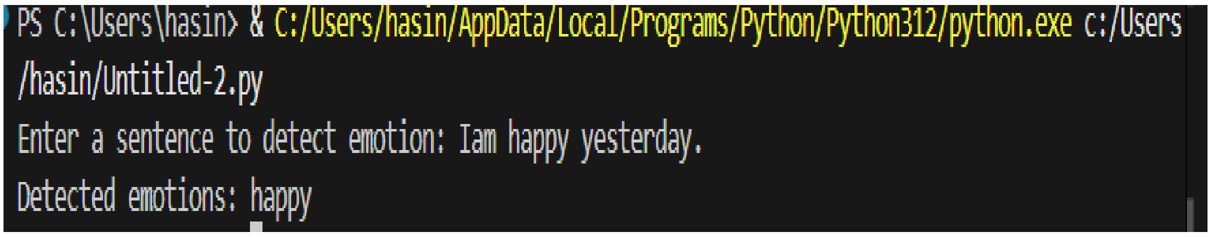
emotions = ['happy', 'sad', 'angry', 'excited', 'nervous', 'neutral']

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:**



```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter a sentence to detect emotion: Iam happy yesterday.
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

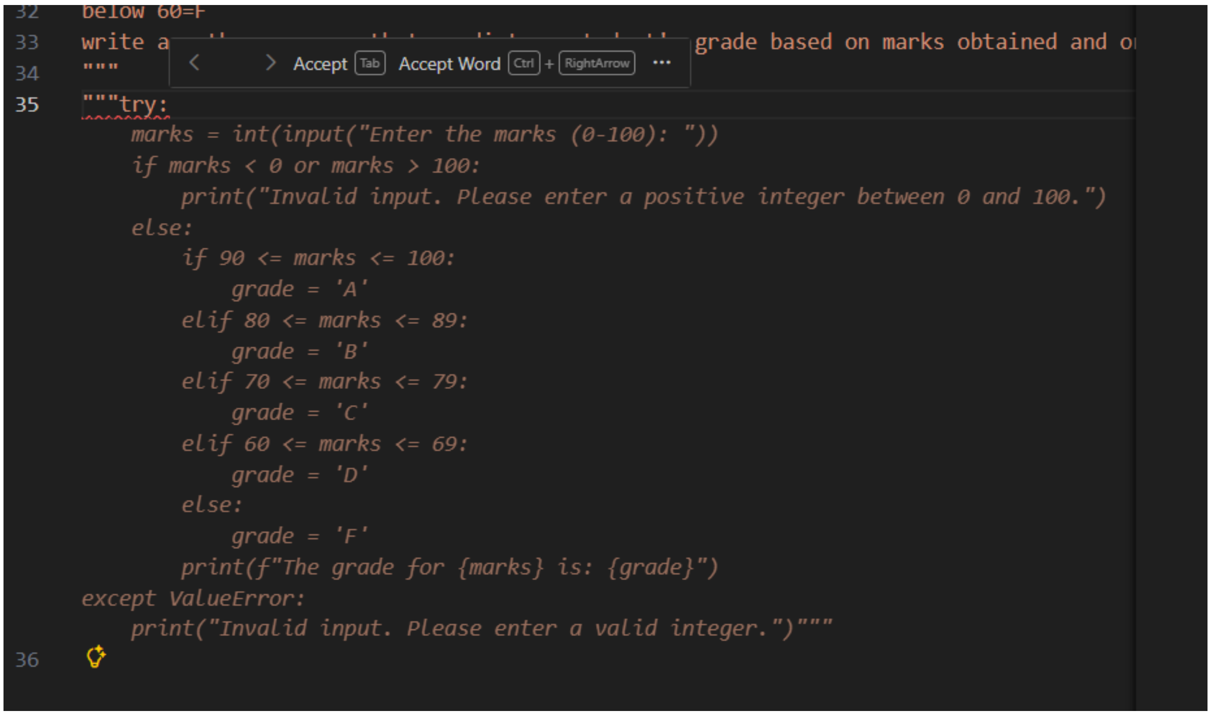
80-89=B

70-79=C

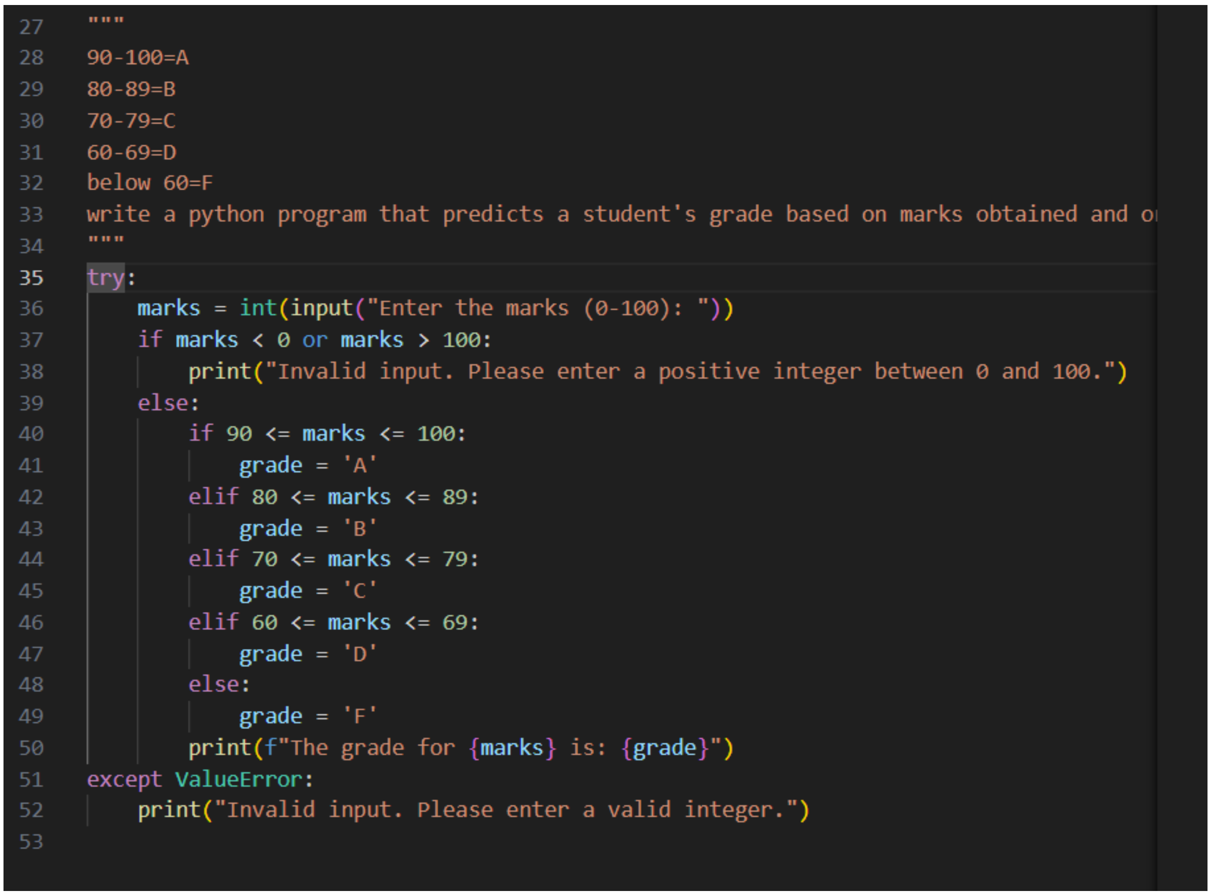
60-69=D

below 60=F

write a python program that predicts a student's grade based on marks obtained and only accept positive integer values from the user otherwise display invalid input.



```
below 60=F
write a                                              grade based on marks obtained and o
"""
"""try:
    marks = int(input("Enter the marks (0-100): "))
    if marks < 0 or marks > 100:
        print("Invalid input. Please enter a positive integer between 0 and 100.")
    else:
        if 90 <= marks <= 100:
            grade = 'A'
        elif 80 <= marks <= 89:
            grade = 'B'
        elif 70 <= marks <= 79:
            grade = 'C'
        elif 60 <= marks <= 69:
            grade = 'D'
        else:
            grade = 'F'
        print(f"The grade for {marks} is: {grade}")
except ValueError:
    print("Invalid input. Please enter a valid integer.")"""
```

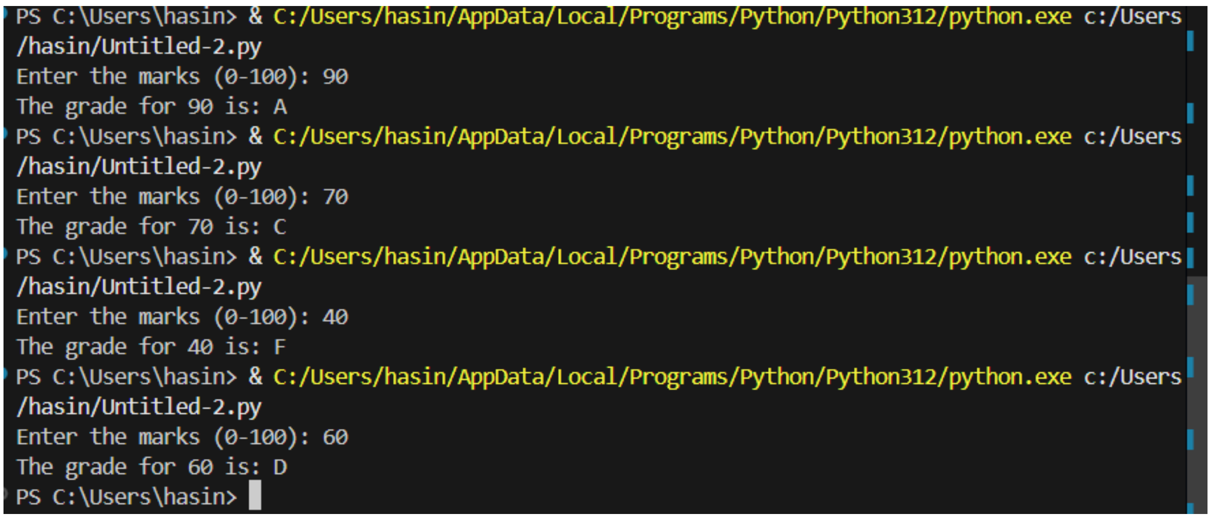


```
"""
90-100=A
80-89=B
70-79=C
60-69=D
below 60=F
write a python program that predicts a student's grade based on marks obtained and o
"""
try:
    marks = int(input("Enter the marks (0-100): "))
    if marks < 0 or marks > 100:
        print("Invalid input. Please enter a positive integer between 0 and 100.")
    else:
        if 90 <= marks <= 100:
            grade = 'A'
        elif 80 <= marks <= 89:
            grade = 'B'
        elif 70 <= marks <= 79:
            grade = 'C'
        elif 60 <= marks <= 69:
            grade = 'D'
        else:
            grade = 'F'
        print(f"The grade for {marks} is: {grade}")
except ValueError:
    print("Invalid input. Please enter a valid integer.")
```

Code:

```
try:
    marks = int(input("Enter the marks (0-100): "))
    if marks < 0 or marks > 100:
        print("Invalid input. Please enter a positive integer between 0 and 100.")
    else:
        if 90 <= marks <= 100:
            grade = 'A'
        elif 80 <= marks <= 89:
            grade = 'B'
        elif 70 <= marks <= 79:
            grade = 'C'
        elif 60 <= marks <= 69:
            grade = 'D'
        else:
            grade = 'F'
        print(f"The grade for {marks} is: {grade}")
except ValueError:
    print("Invalid input. Please enter a valid integer.")
```

Inputs/outputs:



```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter the marks (0-100): 90
The grade for 90 is: A
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter the marks (0-100): 70
The grade for 70 is: C
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter the marks (0-100): 40
The grade for 40 is: F
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter the marks (0-100): 60
The grade for 60 is: D
PS C:\Users\hasin>
```

## 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

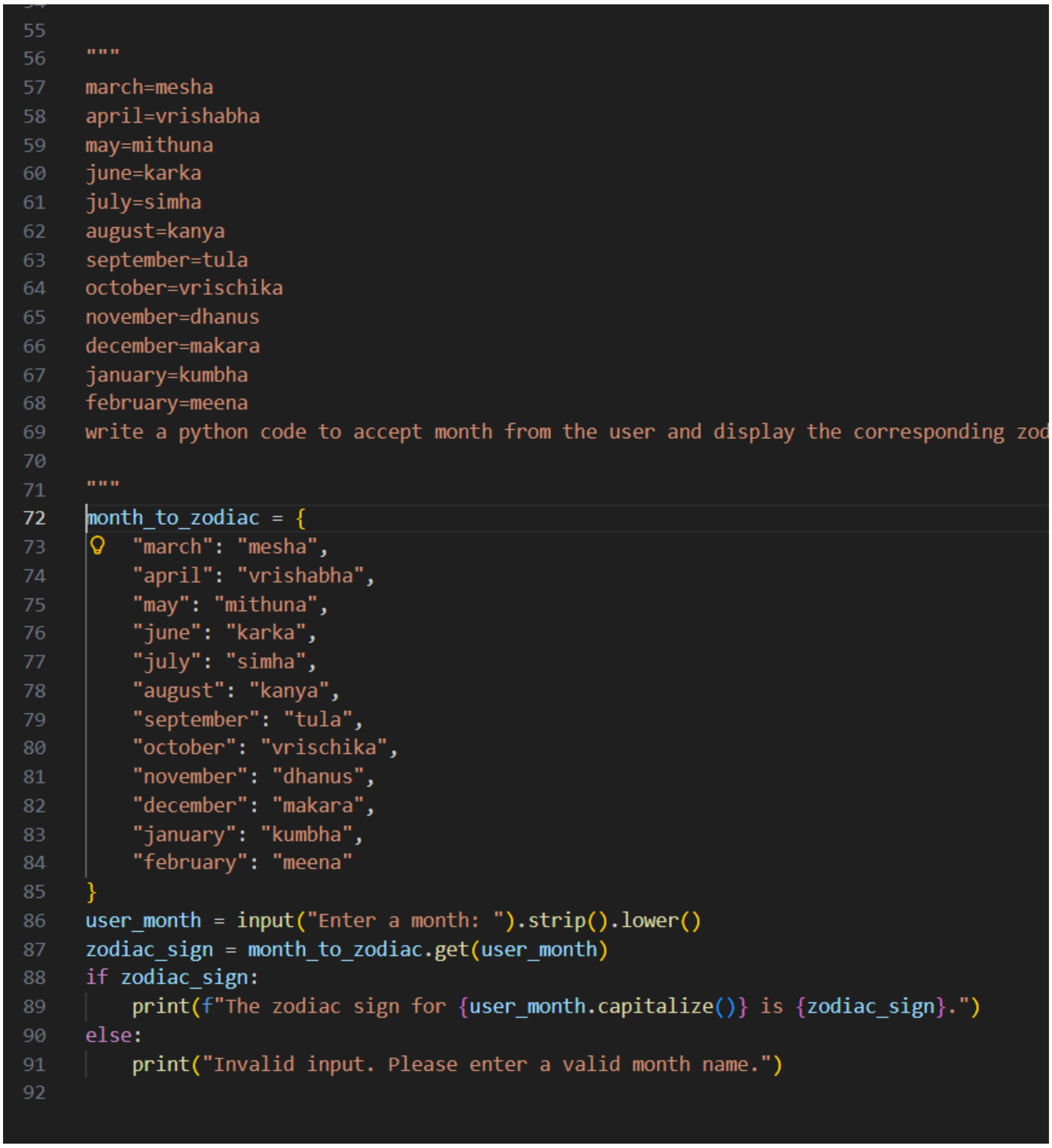
november=dhanus

december=makara

january=kumbha

february=meena

write a python code to accept month from the user and display the corresponding zodiac sign and only accept valid month names otherwise display invalid input



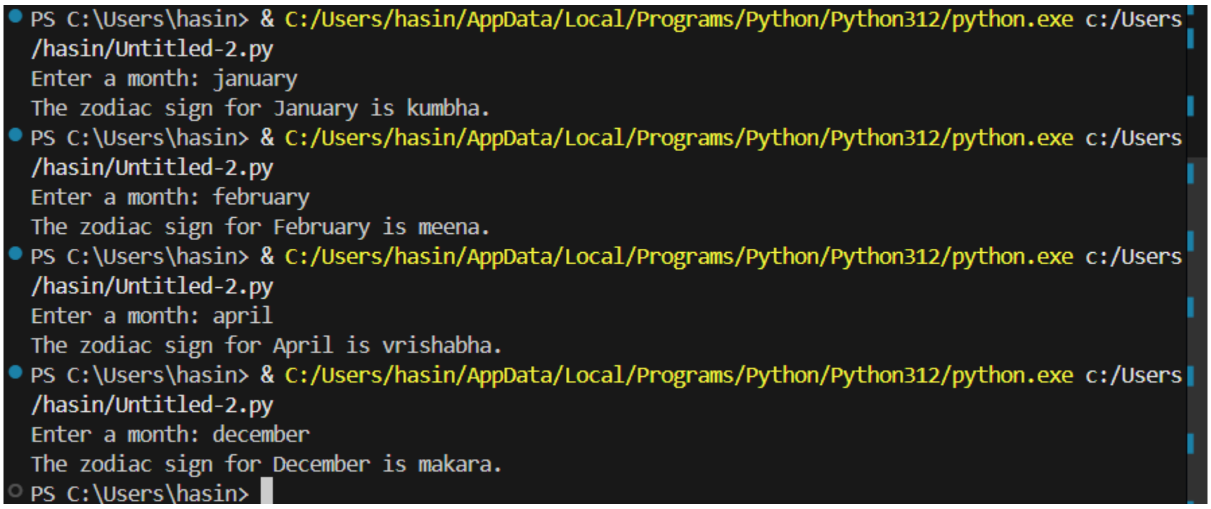
```
"""
march=mesha
april=vrishabha
may=mithuna
june=karka
july=simha
august=kanya
september=tula
october=vrischika
november=dhanus
december=makara
january=kumbha
february=meena
write a python code to accept month from the user and display the corresponding zod

"""
month_to_zodiac = {
    "march": "mesha",
    "april": "vrishabha",
    "may": "mithuna",
    "june": "karka",
    "july": "simha",
    "august": "kanya",
    "september": "tula",
    "october": "vrischika",
    "november": "dhanus",
    "december": "makara",
    "january": "kumbha",
    "february": "meena"
}
user_month = input("Enter a month: ").strip().lower()
zodiac_sign = month_to_zodiac.get(user_month)
if zodiac_sign:
    print(f"The zodiac sign for {user_month.capitalize()} is {zodiac_sign}.")
else:
    print("Invalid input. Please enter a valid month name.")
```

Code:

```
month_to_zodiac = {
    "march": "mesha",
    "april": "vrishabha",
    "may": "mithuna",
    "june": "karka",
    "july": "simha",
    "august": "kanya",
    "september": "tula",
    "october": "vrischika",
    "november": "dhanus",
"december": "makara",
    "january": "kumbha",
    "february": "meena"
}
user_month = input("Enter a month: ").strip().lower()
zodiac_sign = month_to_zodiac.get(user_month)
if zodiac_sign:
    print(f"The zodiac sign for {user_month.capitalize()} is {zodiac_sign}.")
else:
    print("Invalid input. Please enter a valid month name.")
```

Inputs/outputs:



```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter a month: january
The zodiac sign for January is kumbha.
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter a month: february
The zodiac sign for February is meena.
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter a month: april
The zodiac sign for April is vrishabha.
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter a month: december
The zodiac sign for December is makara.
PS C:\Users\hasin>
```

**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:

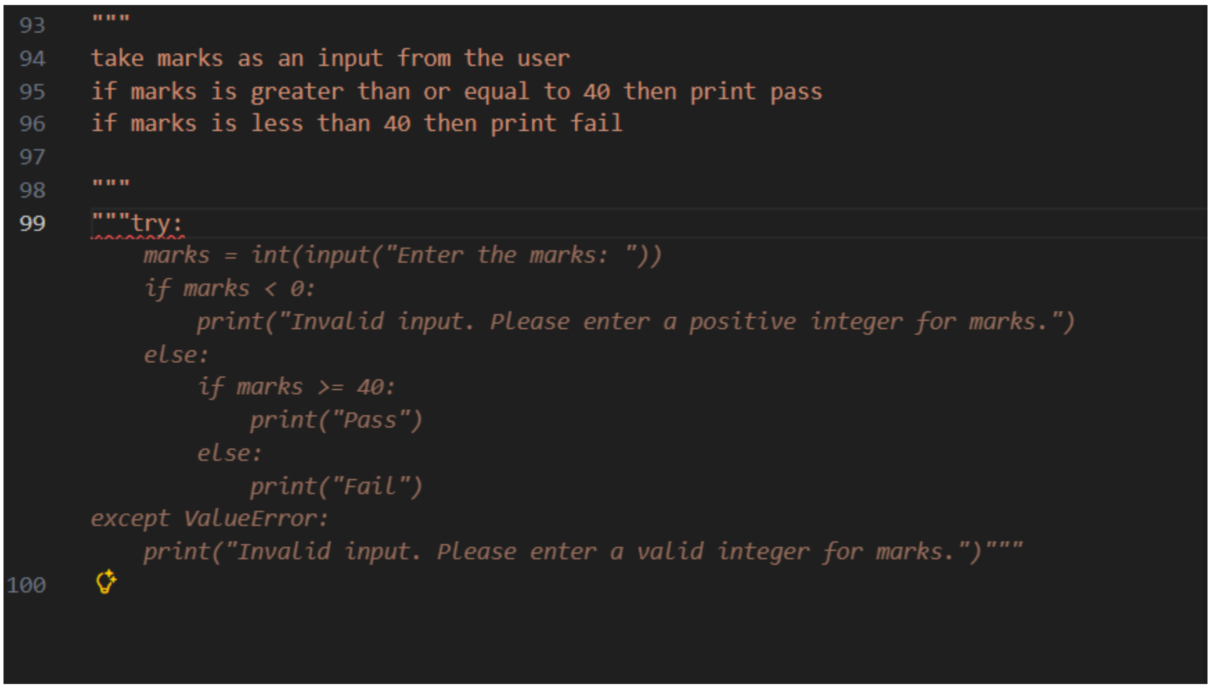
['Pass', 'Fail']

Prompt:

take marks as an input from the user

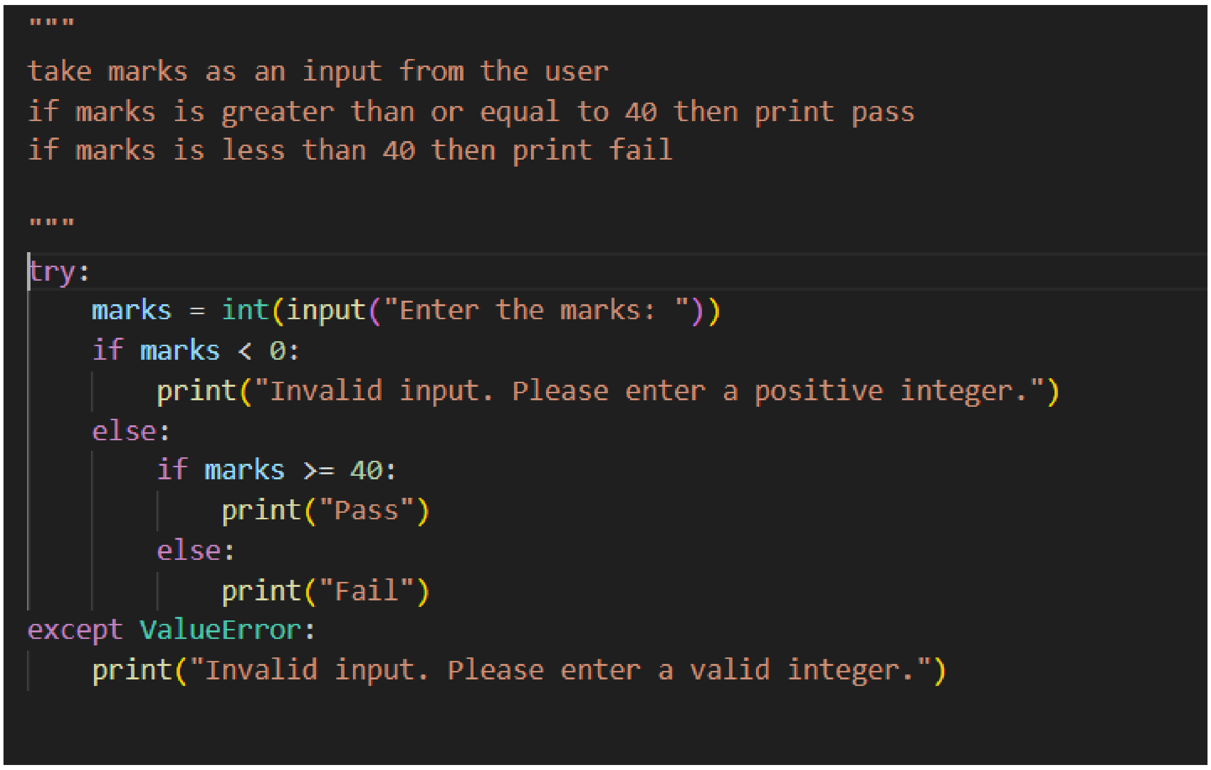
if marks is greater than or equal to 40 then print pass

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.")"""
```



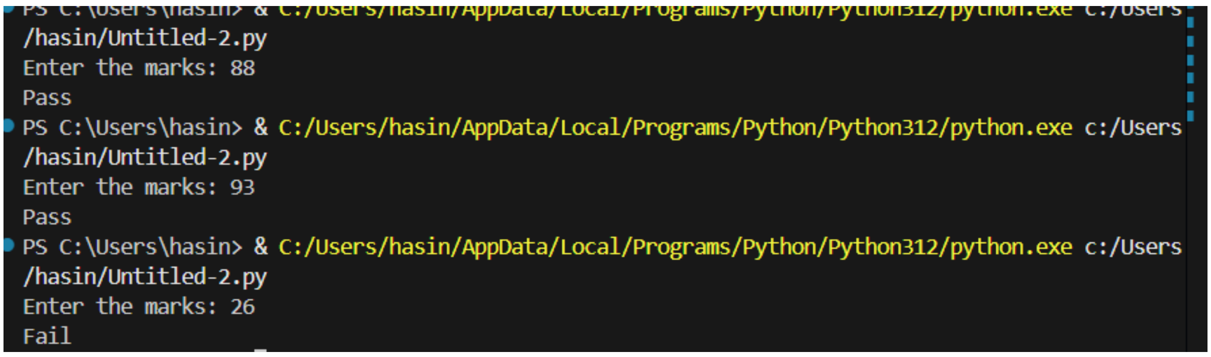
```
"""
take marks as an input from the user
if marks is greater than or equal to 40 then print pass
if marks is less than 40 then print fail

"""
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.")
```

Inputs/outputs:



```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter the marks: 88
Pass
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter the marks: 93
Pass
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter the marks: 26
Fail
```

**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:

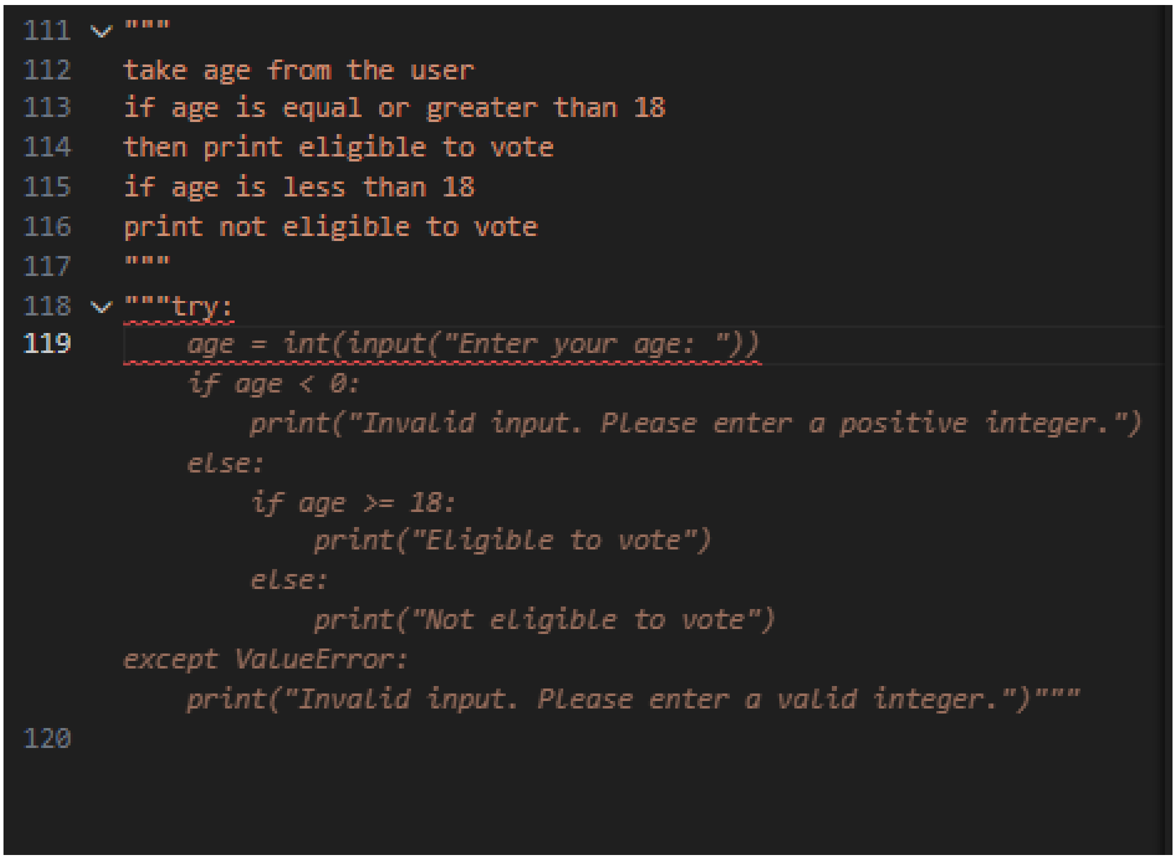
take age from the user

if age is equal or greater than 18

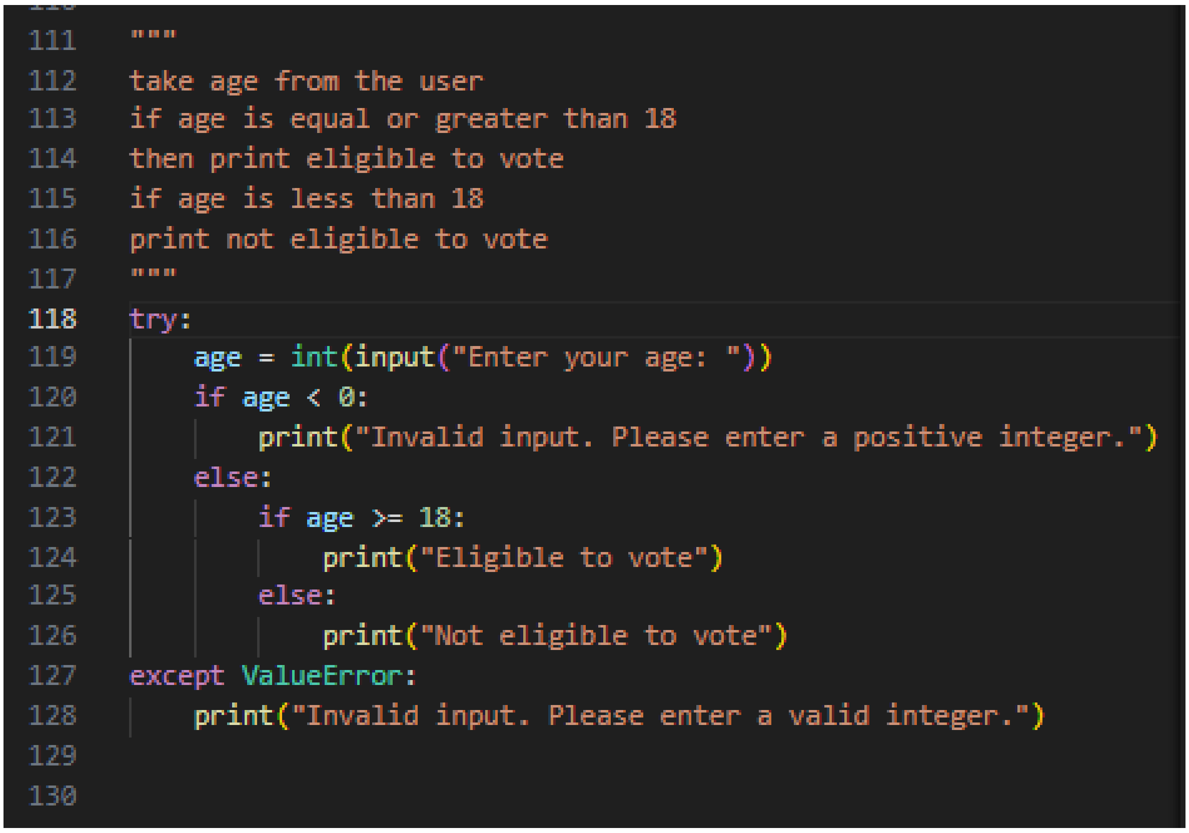
then print eligible to vote

if age is less than 18

print not eligible to vote



```
"""
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
"""
"""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.")"""
```

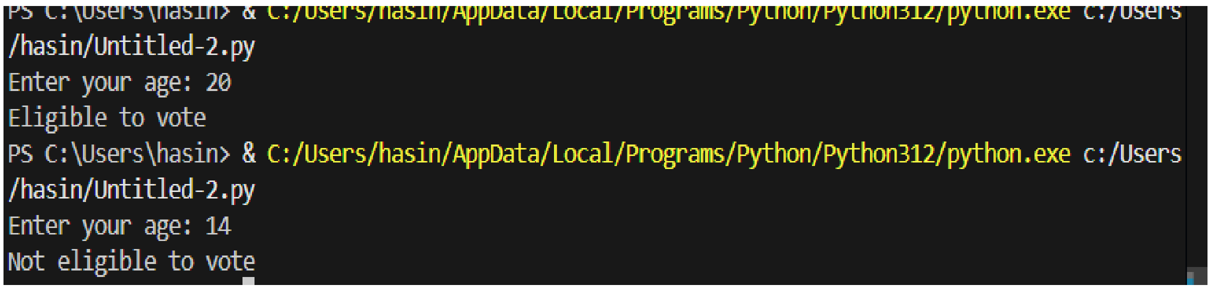


```
"""
take age from the user
if age is equal or greater than 18
then print eligible to vote
if age is less than 18
print not eligible to vote
"""
try:
    age = int(input("Enter your age: "))
    if age < 0:
        print("Invalid input. Please enter a positive integer.")
    else:
        if age >= 18:
            print("Eligible to vote")
        else:
            print("Not eligible to vote")
except ValueError:
    print("Invalid input. Please enter a valid integer.")
```

Code:

```
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.")
```

Inputs/outputs:



```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter your age: 20
Eligible to vote
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter your age: 14
Not eligible to vote
```

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

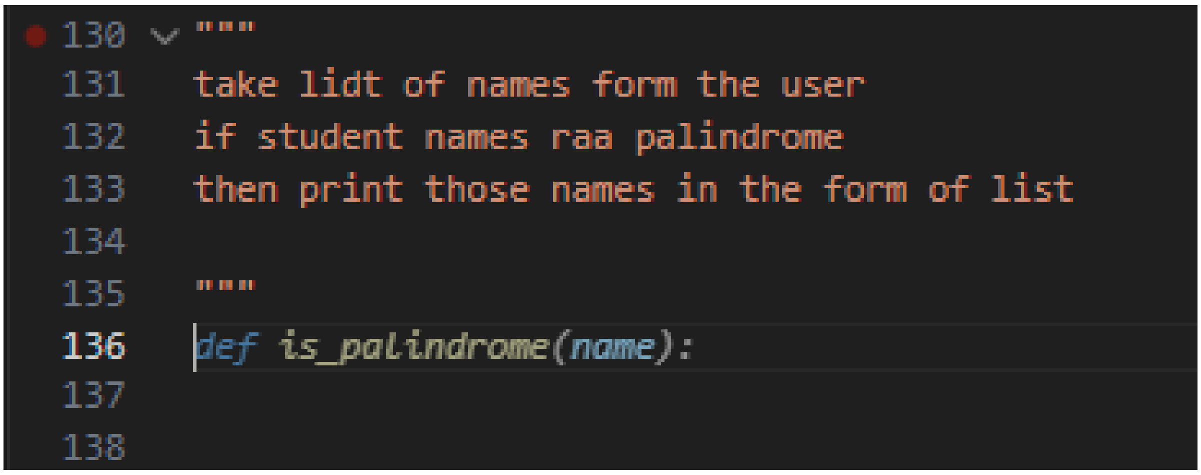
Task: Write a Python program that uses the prompt chaining technique to identify palindrome names from a list of student names.

Prompt:

take list of names form the user

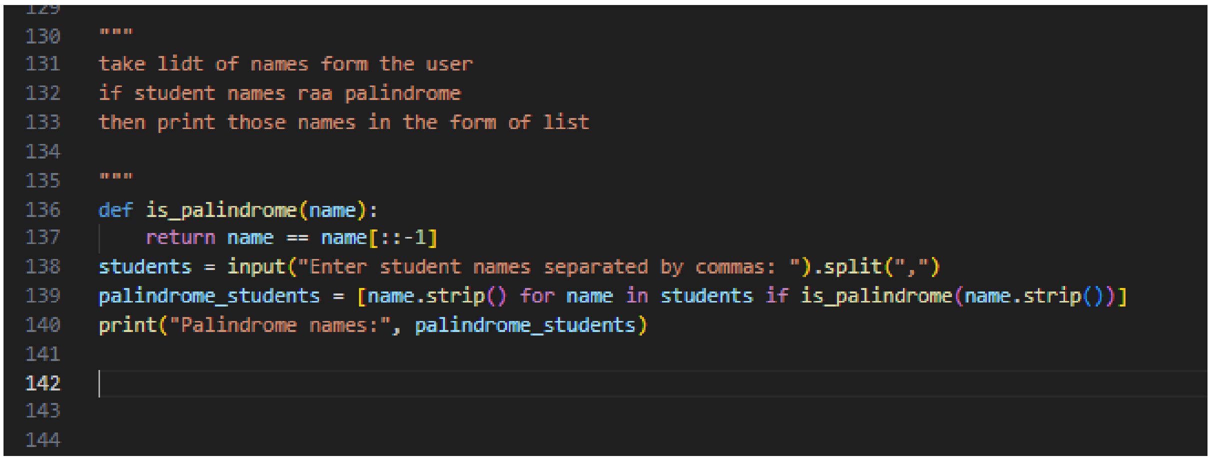
if student names raa palindrome

then print those names in the form of list



```
"""
take lidt of names form the user
if student names raa palindrome
then print those names in the form of list

"""
def is_palindrome(name):
```



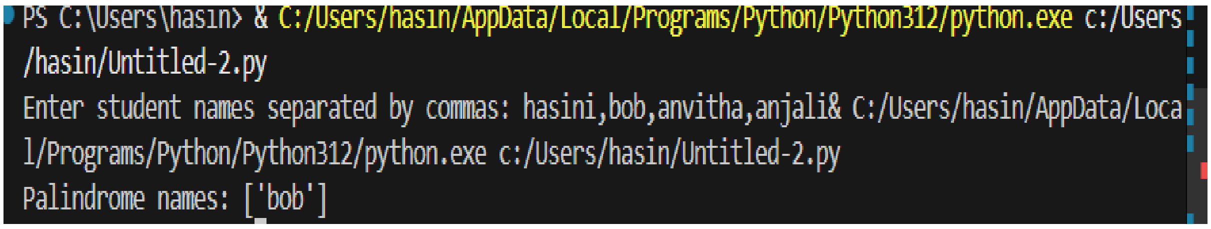
```
"""
take lidt of names form the user
if student names raa palindrome
then print those names in the form of list

"""
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)
```

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)
```

Inputs/outputs:



```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Programs/Python/Python312/python.exe c:/Users
/hasin/Untitled-2.py
Enter student names separated by commas: hasini,bob,anvitha,anjali& C:/Users/hasin/AppData/Loca
l/Programs/Python/Python312/python.exe c:/Users/hasin/Untitled-2.py
Palindrome names: ['bob']
```

**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

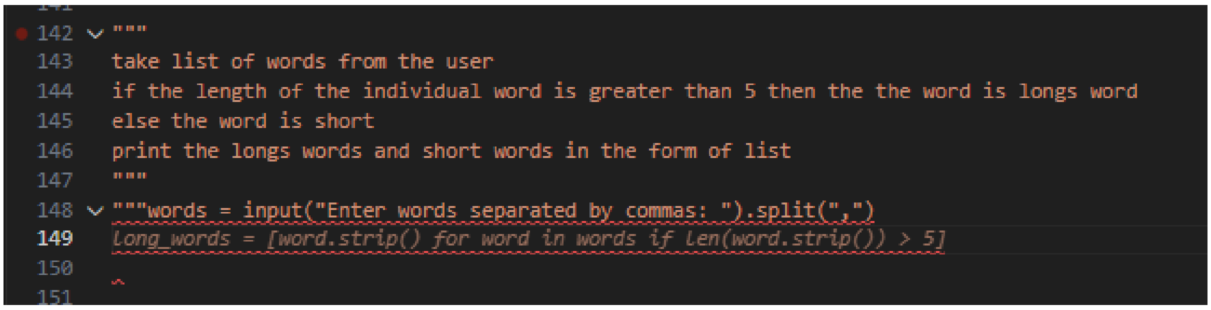
Prompt:

take list of words from the user

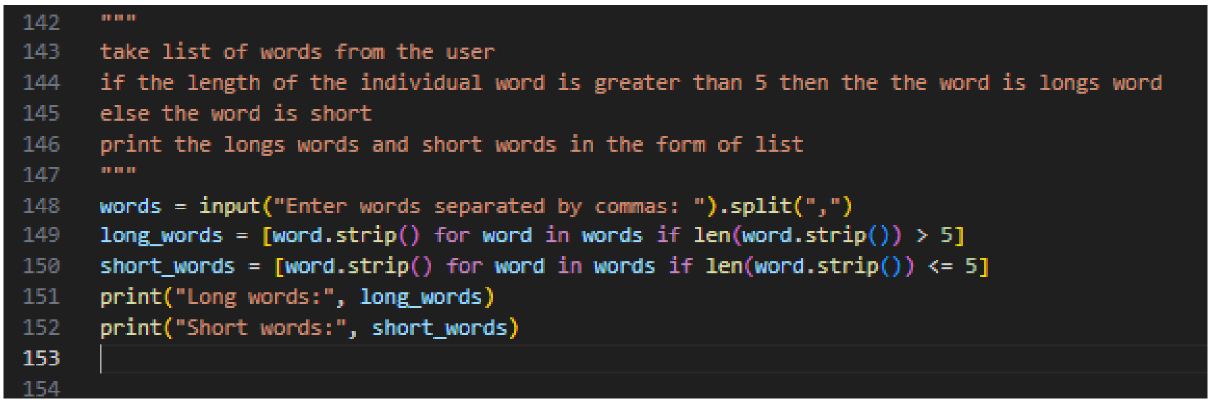
if the length of the individual word is greater than 5 then the the word is longs word

else the word is short

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

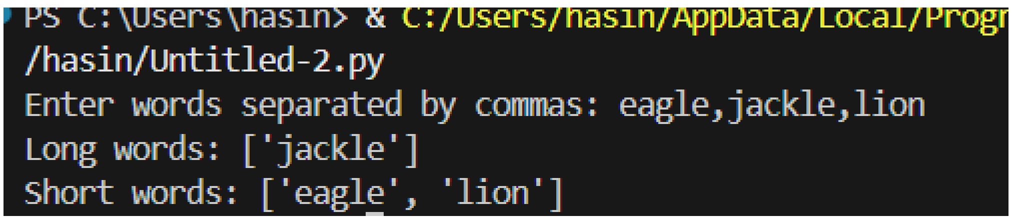


```python
"""
take list of words from the user
if the length of the individual word is greater than 5 then the the word is longs word
else the word is short
print the longs words and short words in the form of list
"""
"""words = input("Enter words separated by commas: ").split(",")
long_words = [word.strip() for word in words if len(word.strip()) > 5]
```



```python
"""
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
"""
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)
```

Inputs/outputs:



```
PS C:\Users\hasin> & C:/Users/hasin/AppData/Local/Progr
/hasin/Untitled-2.py
Enter words separated by commas: eagle,jackle,lion
Long words: ['jackle']
Short words: ['eagle', 'lion']
```