Batch-06 **id-2403A510F4** **Name:KOLA SNEHA**

SCHOOL OF COMPUTERSCIENCEANDARTIFICIAL INTELLIGENCE		**DEPARTMENTOFCOMPUTERSCIENCE ENGINEERING**
ProgramName:B.Tech	**AssignmentType:Lab**	**AcademicYear:**2025-2026
CourseCoordinatorName	VenkataramanaVeeramsetty	
Instructor(s)Name	Dr.V.Venkataramana(Co-ordinator) Dr.T.SampathKumar Dr.PramodaPatro Dr.BrijKishor Tiwari Dr.J.Ravichander Dr.MohammandAliShaik Dr.AnirodhKumar Mr.S.Naresh Kumar Dr.RAJESHVELPULA Mr.KundhanKumar Ms.Ch.Rajitha Mr.MPrakash Mr.B.Raju Intern1(Dharmateja) Intern2(Sai Prasad) Intern3(Sowmya) NS_2(Mounika)	

CourseCode	24CS002PC215	**CourseTitle**	AIAssistedCoding
Year/Sem	II/I	**Regulation**	R24
Date and Day ofAssignment	Week2 - Wednesday	**Time(s)**	
Duration	2Hours	**Applicableto Batches**	

AssignmentNumber:2.3(Presentassignmentnumber)/**24**(Totalnumberofassignments)

Q.No.	Question	ExpectedTime to complete
1	Lab4:AdvancedPromptEngineering – Zero-shot, One-shot,andFew-shotTechniques **Lab Objectives:** - ToexploreandapplydifferentlevelsofpromptexamplesinAI-assistedcode generation.	Week2 - Wednesday

- Tounderstandhowzero-shot,one-shot,andfew-shotpromptingaffectAIoutput quality.
- Toevaluatetheimpactofcontextrichnessandexample quantityonAI performance.
- Tobuild awarenessofpromptstrategy effectivenessfordifferentproblem types.

LabOutcomes (LOs):

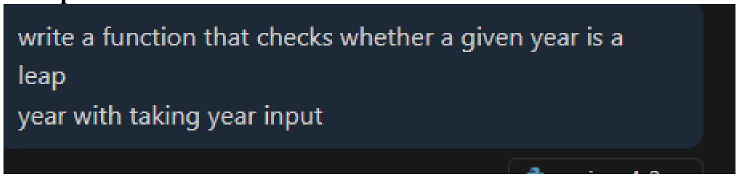
Aftercompleting thislab,studentswill beableto:

- Usezero-shotpromptingtoinstructAIwithminimalcontext.
- Useone-shotpromptingwith asingle exampletoguideAIcodegeneration.
- Applyfew-shotpromptingusingmultiple examplesto improveAIresponses.
- CompareAIoutputsacrossthethreeprompting strategies.

TaskDescription#1

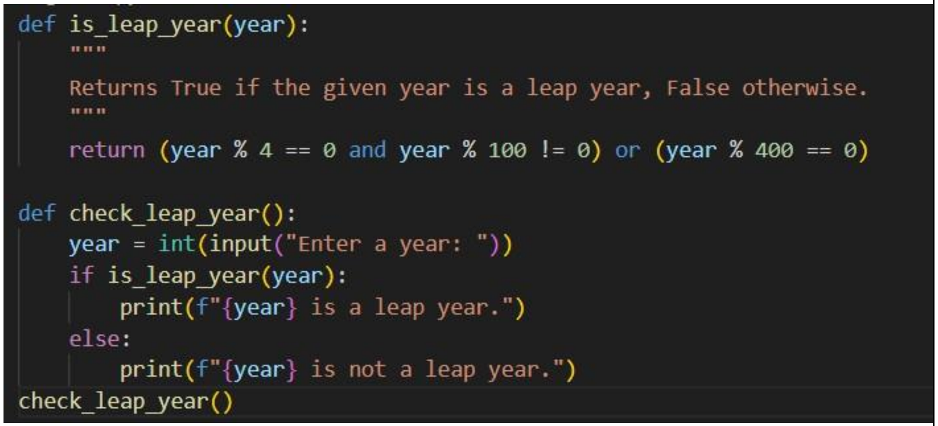
- Zero-shot:PromptAItowriteafunctionthatcheckswhetheragivenyearisaleap year.

Prompt:



write a function that checks whether a given year is a
leap
year with taking year input

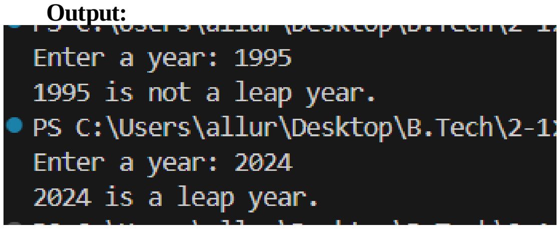
Code:



```
def is_leap_year(year):
    """
    Returns True if the given year is a leap year, False otherwise.
    """
    return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)

def check_leap_year():
    year = int(input("Enter a year: "))
    if is_leap_year(year):
        print(f"{year} is a leap year.")
    else:
        print(f"{year} is not a leap year.")
check_leap_year()
```

Output:



```
Enter a year: 1995
1995 is not a leap year.
PS C:\Users\allur\Desktop\B.Tech\2-1
Enter a year: 2024
2024 is a leap year.
```

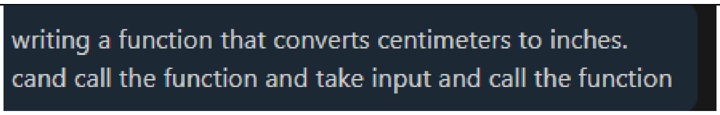
ExpectedOutput#1

- AI-generatedfunctionwithnoexamplesprovided

TaskDescription#2

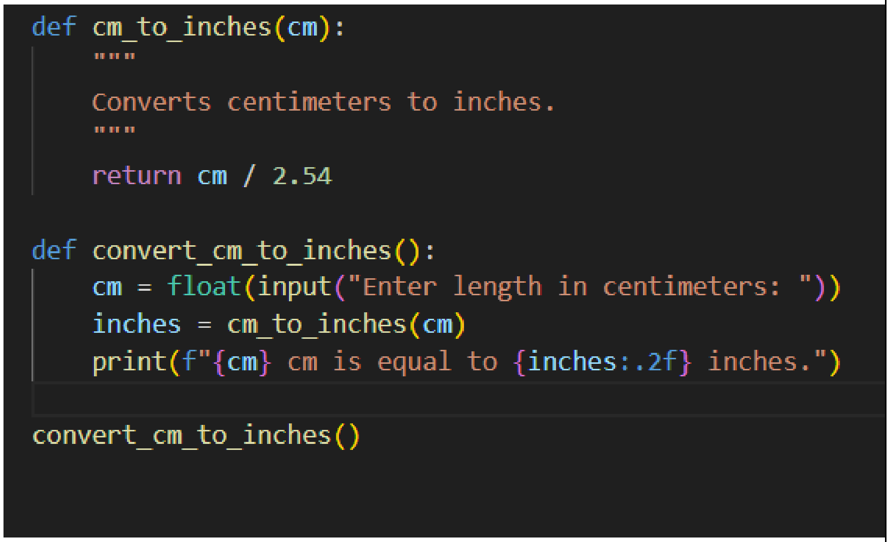
- One-shot:Giveoneinput-outputexampletoguideAIinwritingafunctionthat converts centimeters to inches.

Prompt:



writing a function that converts centimeters to inches.
cand call the function and take input and call the function

Code:

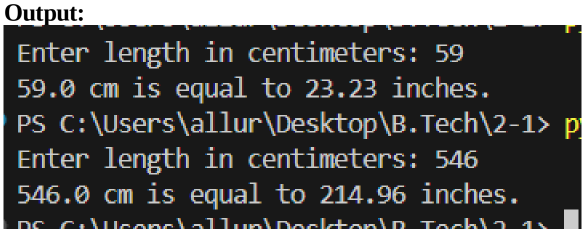


```python
def cm_to_inches(cm):
    """
    Converts centimeters to inches.
    """
    return cm / 2.54

def convert_cm_to_inches():
    cm = float(input("Enter length in centimeters: "))
    inches = cm_to_inches(cm)
    print(f"{cm} cm is equal to {inches:.2f} inches.")

convert_cm_to_inches()
```

Output:



```
Enter length in centimeters: 59
59.0 cm is equal to 23.23 inches.
PS C:\Users\allur\Desktop\B.Tech\2-1> p
Enter length in centimeters: 546
546.0 cm is equal to 214.96 inches.
```

ExpectedOutput#2

- Functionwithcorrectconversionlogic

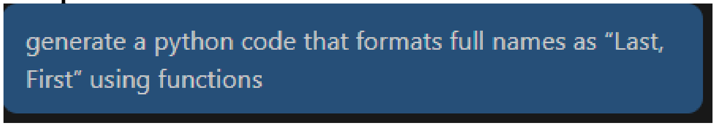
TaskDescription#3

- Few-shot:Provide2–3 examplestogenerateafunctionthatformatsfullnamesas “Last,First”.

ExpectedOutput#3

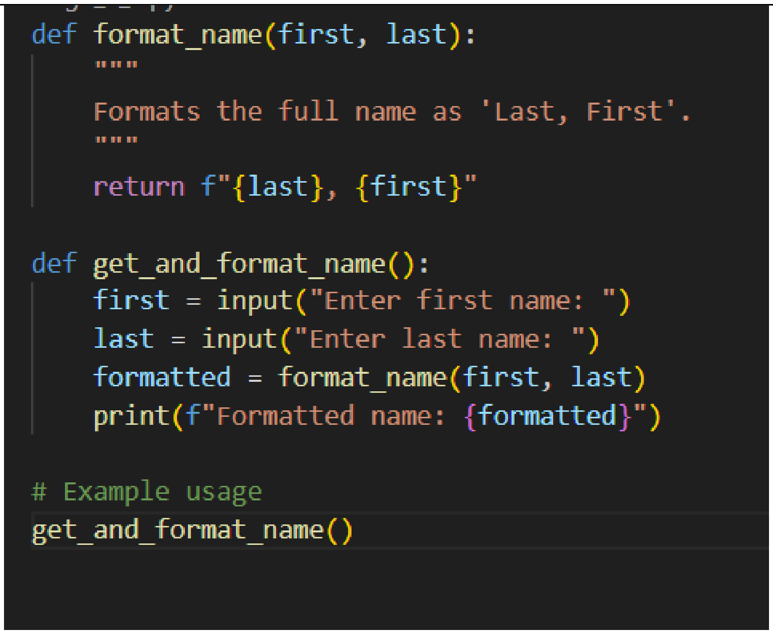
- Well-structuredfunctionrespectingthe examples

Prompt:



generate a python code that formats full names as “Last, First” using functions

Code:

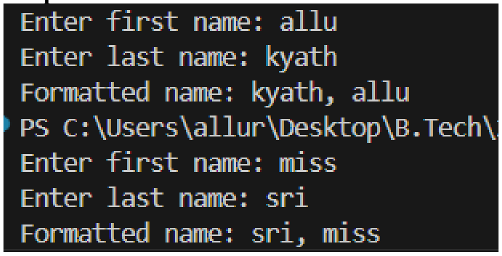


```python
def format_name(first, last):
    """
    Formats the full name as 'Last, First'.
    """
    return f"{last}, {first}"

def get_and_format_name():
    first = input("Enter first name: ")
    last = input("Enter last name: ")
    formatted = format_name(first, last)
    print(f"Formatted name: {formatted}")

# Example usage
get_and_format_name()
```

Output:



```
Enter first name: allu
Enter last name: kyath
Formatted name: kyath, allu
PS C:\Users\allur\Desktop\B.Tech\
Enter first name: miss
Enter last name: sri
Formatted name: sri, miss
```

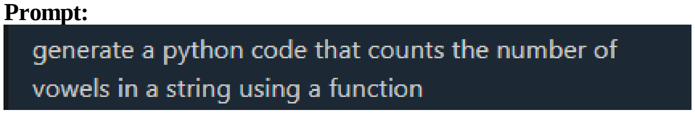
TaskDescription#4

- Comparezero-shotandfew-shotpromptsforwritingafunctionthatcountsthe number of vowels in a string.

ExpectedOutput#4

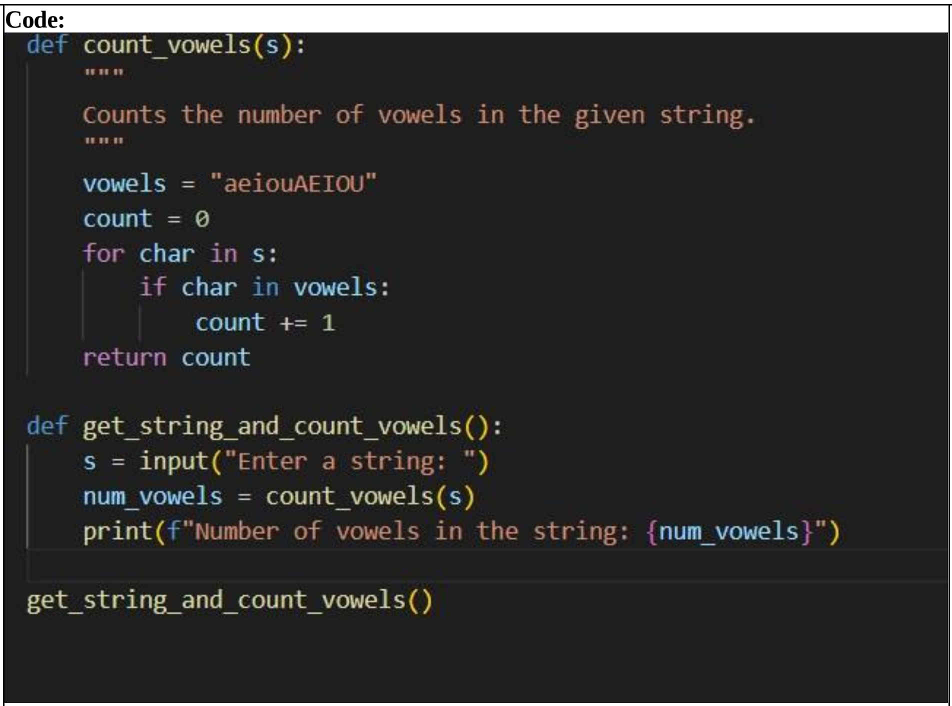
- Functionaloutputandcomparativereflection

Prompt:



generate a python code that counts the number of vowels in a string using a function

Code:

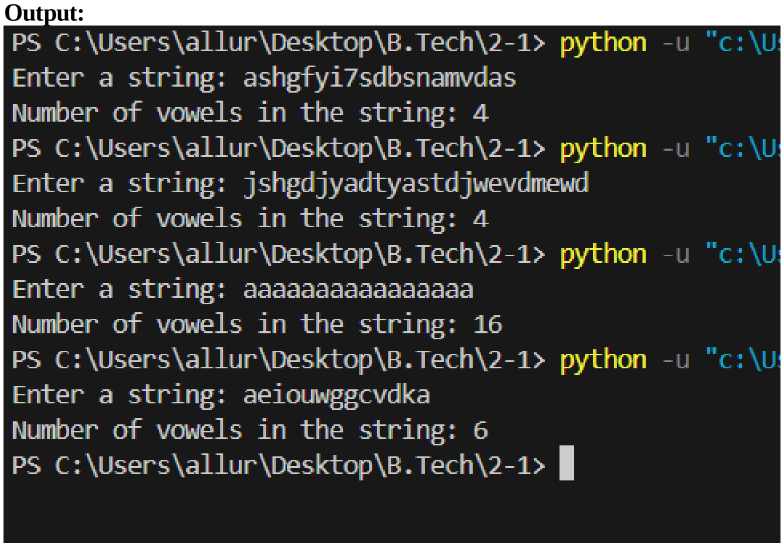


```
def count_vowels(s):
    """
    Counts the number of vowels in the given string.
    """
    vowels = "aeiouAEIOU"
    count = 0
    for char in s:
        if char in vowels:
            count += 1
    return count

def get_string_and_count_vowels():
    s = input("Enter a string: ")
    num_vowels = count_vowels(s)
    print(f"Number of vowels in the string: {num_vowels}")

get_string_and_count_vowels()
```

Output:



```
PS C:\Users\allur\Desktop\B.Tech\2-1> python -u "c:\U
Enter a string: ashgfyi7sdbsnamvdas
Number of vowels in the string: 4
PS C:\Users\allur\Desktop\B.Tech\2-1> python -u "c:\U
Enter a string: jshgdjyadtyastdjwevdmewd
Number of vowels in the string: 4
PS C:\Users\allur\Desktop\B.Tech\2-1> python -u "c:\U
Enter a string: aaaaaaaaaaaaaaaa
Number of vowels in the string: 16
PS C:\Users\allur\Desktop\B.Tech\2-1> python -u "c:\U
Enter a string: aeiouwggcvdka
Number of vowels in the string: 6
PS C:\Users\allur\Desktop\B.Tech\2-1> 
```

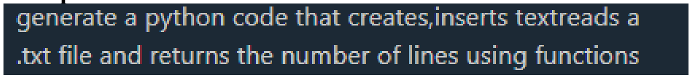
TaskDescription#5

- Usefew-shotpromptingtogenerateafunctionthatreadsa.txtfileandreturnsthe number of lines.

ExpectedOutput#5

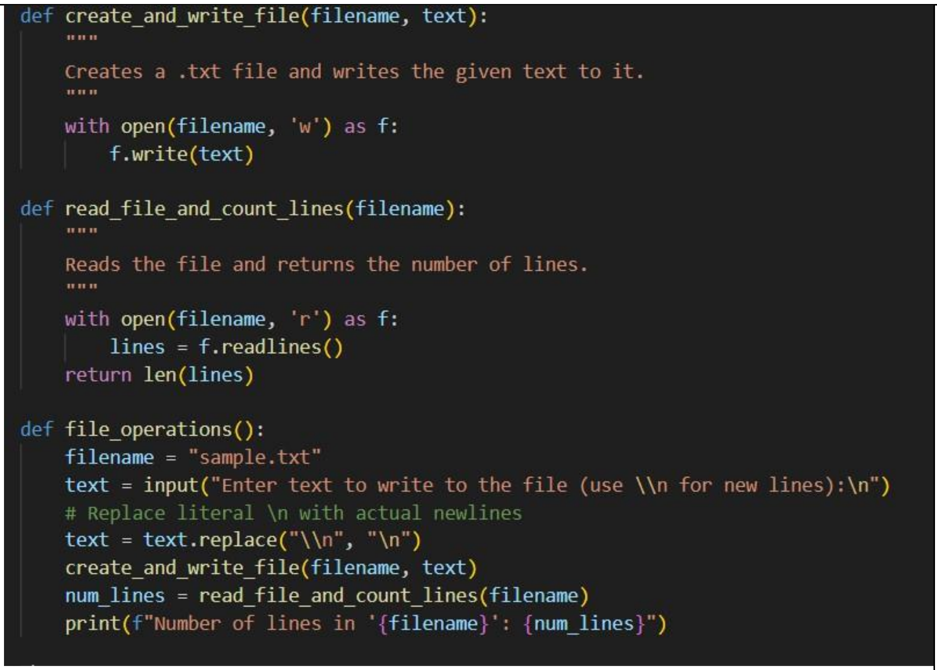
- Workingfile-processingfunctionwithAI-guided logic

Prompt:



generate a python code that creates,inserts textreads a .txt file and returns the number of lines using functions

Code:

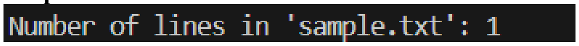


```python
def create_and_write_file(filename, text):
    """
    Creates a .txt file and writes the given text to it.
    """
    with open(filename, 'w') as f:
        f.write(text)

def read_file_and_count_lines(filename):
    """
    Reads the file and returns the number of lines.
    """
    with open(filename, 'r') as f:
        lines = f.readlines()
    return len(lines)

def file_operations():
    filename = "sample.txt"
    text = input("Enter text to write to the file (use \\n for new lines):\n")
    # Replace literal \n with actual newlines
    text = text.replace("\\n", "\n")
    create_and_write_file(filename, text)
    num_lines = read_file_and_count_lines(filename)
    print(f"Number of lines in '{filename}': {num_lines}")
```

Output:



```
Number of lines in 'sample.txt': 1
```

Note:Reportshouldbesubmittedaworddocumentforalltasksinasingledocumentwith prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

Criteria	**Max Marks**
ZeroShot(Task#1)	0.5
OneShot (Task#2)	0.5
FewShot(Task#3&Task#5)	1.0
Comparison(Task#4)	0.5
Total	**2.5 Marks**