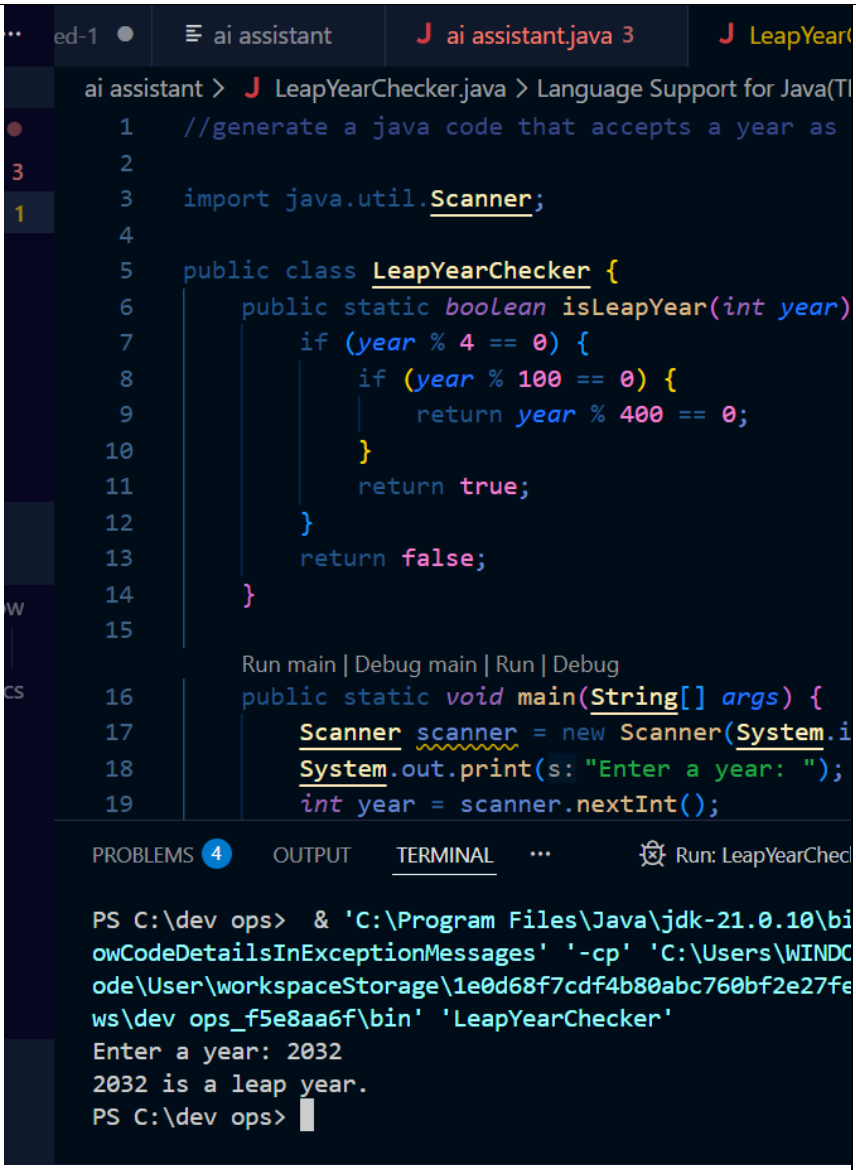
Assignment Number: 4.3

NAME:M.AKASH	ROLL NO:2303A51820	BATCH:26

Q.No.	Question	*Expected Time to complete*
1	**Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques** **Lab Objectives** • To explore and apply different levels of prompt examples in AI-assisted code generation • To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality • To evaluate the impact of context richness and example quantity on AI performance • To build awareness of prompt strategy effectiveness for different problem types **Lab Outcomes (LOs)** **After completing this lab, students will be able to:** • Use zero-shot prompting to instruct AI with minimal context • Use one-shot prompting with a single example to guide AI code generation • Apply few-shot prompting using multiple examples to improve AI responses • Compare AI outputs across different prompting strategies	Week2 - Wednesday
	Task 1: Zero-Shot Prompting – Leap Year Check **Scenario** Zero-shot prompting involves giving instructions without providing examples. **Task Description** Use zero-shot prompting to instruct an AI tool to generate a Python function that: • Accepts a year as input • Checks whether the given year is a leap year • Returns an appropriate result **Note:** No input-output examples should be provided in the prompt. **Expected Output**	



ed-1
ai assistant
ai assistant.java 3
LeapYear(
ai assistant > LeapYearChecker.java > Language Support for Java(Tl
Run main | Debug main | Run | Debug
PROBLEMS 4
OUTPUT
TERMINAL
Run: LeapYearChecl

```java
//generate a java code that accepts a year as

import java.util.Scanner;

public class LeapYearChecker {
    public static boolean isLeapYear(int year)
        if (year % 4 == 0) {
            if (year % 100 == 0) {
                return year % 400 == 0;
            }
            return true;
        }
        return false;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.i
        System.out.print(s: "Enter a year: ");
        int year = scanner.nextInt();
```

```
PS C:\dev ops>  & 'C:\Program Files\Java\jdk-21.0.10\bi
owCodeDetailsInExceptionMessages' '-cp' 'C:\Users\WINDC
ode\User\workspaceStorage\1e0d68f7cdf4b80abc760bf2e27fe
ws\dev ops_f5e8aa6f\bin' 'LeapYearChecker'
Enter a year: 2032
2032 is a leap year.
PS C:\dev ops> 
```

Task 2: One-Shot Prompting – Centimeters to Inches Conversion

Scenario

One-shot prompting guides AI using a single example.

Task Description

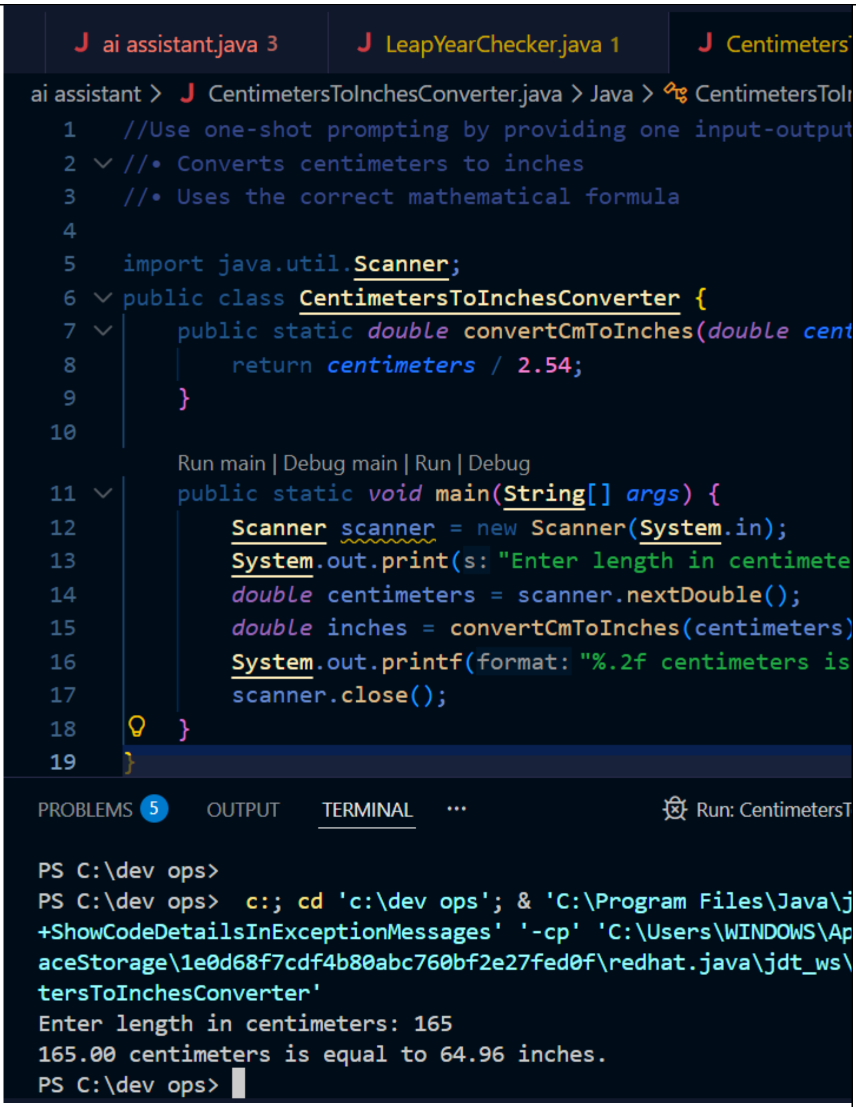
Use one-shot prompting by providing one input-output example to generate a Python function that:

- Converts centimeters to inches
- Uses the correct mathematical formula

Example provided in prompt:

Input: 10 cm → Output: 3.94 inches

Expected Output



```java
//Use one-shot prompting by providing one input-output
//• Converts centimeters to inches
//• Uses the correct mathematical formula

import java.util.Scanner;
public class CentimetersToInchesConverter {
    public static double convertCmToInches(double cent
        return centimeters / 2.54;
    }

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s: "Enter length in centimete
        double centimeters = scanner.nextDouble();
        double inches = convertCmToInches(centimeters)
        System.out.printf(format: "%.2f centimeters is
        scanner.close();
    }
}
```

```
PS C:\dev ops>
PS C:\dev ops>  c:; cd 'c:\dev ops'; & 'C:\Program Files\Java\j
+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\WINDOWS\Ap
aceStorage\1e0d68f7cdf4b80abc760bf2e27fed0f\redhat.java\jdt_ws\
tersToInchesConverter'
Enter length in centimeters: 165
165.00 centimeters is equal to 64.96 inches.
PS C:\dev ops> 
```

Task 3: Few-Shot Prompting – Name Formatting

Scenario

Few-shot prompting improves accuracy by providing multiple examples.

Task Description

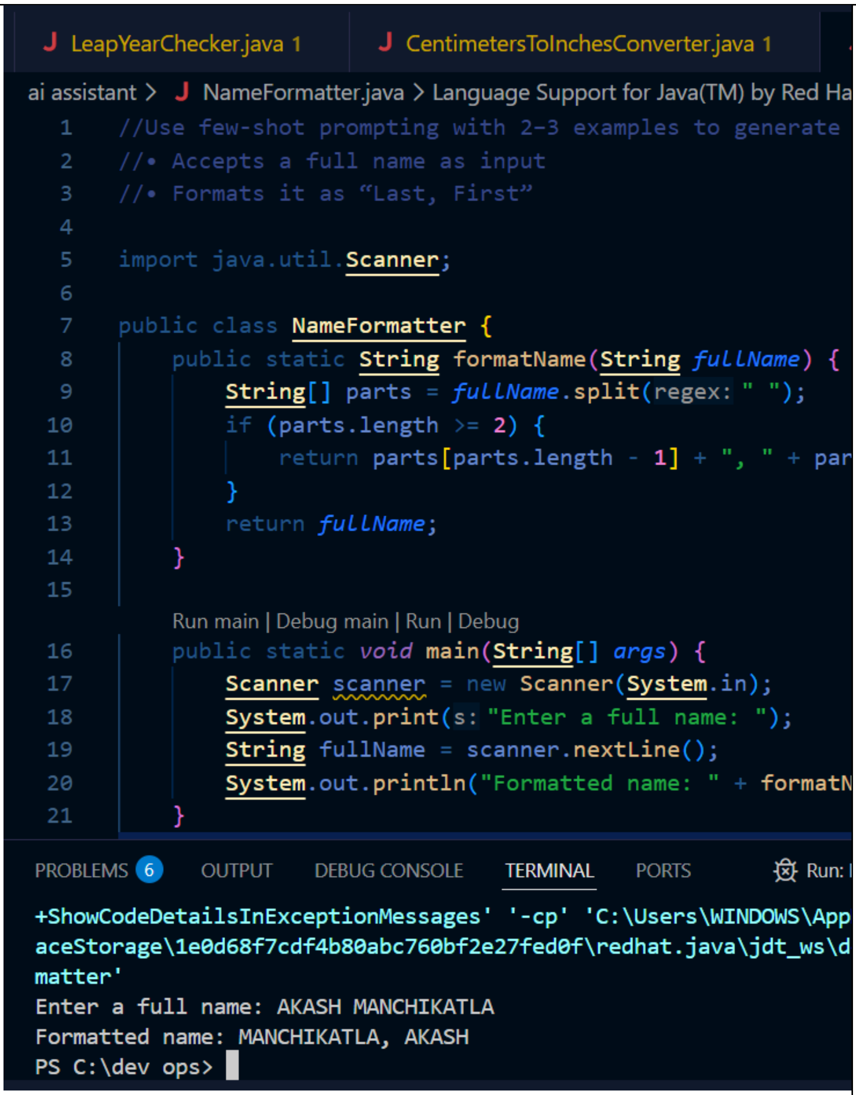
Use few-shot prompting with 2–3 examples to generate a Python function that:

- Accepts a full name as input
- Formats it as “Last, First”

Example formats:

- "John Smith" → "Smith, John"
- "Anita Rao" → "Rao, Anita"

Expected Output


LeapYearChecker.java 1
CentimetersToInchesConverter.java 1
ai assistant > NameFormatter.java > Language Support for Java(TM) by Red Ha
Run main | Debug main | Run | Debug
PROBLEMS 6
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Run: I

```java
//Use few-shot prompting with 2-3 examples to generate
//• Accepts a full name as input
//• Formats it as "Last, First"

import java.util.Scanner;

public class NameFormatter {
    public static String formatName(String fullName) {
        String[] parts = fullName.split(regex: " ");
        if (parts.length >= 2) {
            return parts[parts.length - 1] + ", " + par
        }
        return fullName;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s: "Enter a full name: ");
        String fullName = scanner.nextLine();
        System.out.println("Formatted name: " + formatN
    }
```

```
+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\WINDOWS\App
aceStorage\1e0d68f7cdf4b80abc760bf2e27fed0f\redhat.java\jdt_ws\d
matter'
Enter a full name: AKASH MANCHIKATLA
Formatted name: MANCHIKATLA, AKASH
PS C:\dev ops>
```

Task 4: Comparative Analysis – Zero-Shot vs Few-Shot

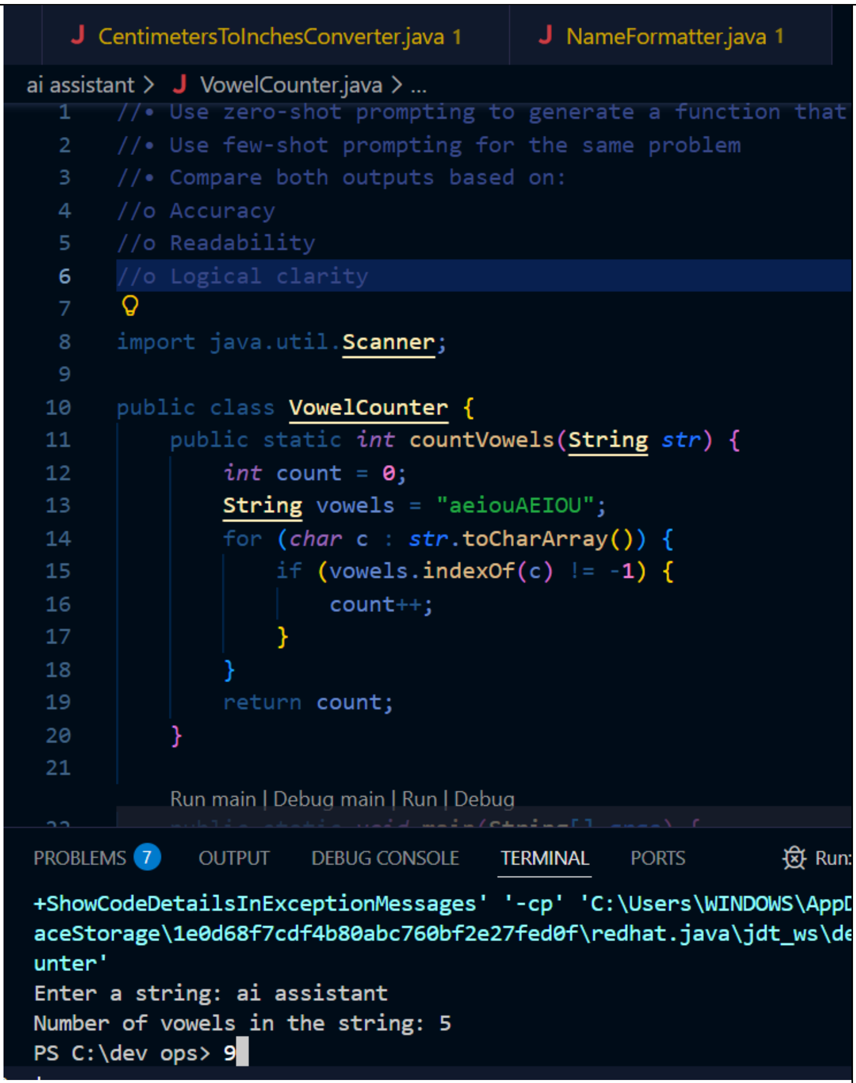
Scenario

Different prompt strategies may produce different code quality.

Task Description

- Use zero-shot prompting to generate a function that counts vowels in a string
- Use few-shot prompting for the same problem
- Compare both outputs based on:
    - Accuracy
    - Readability
    - Logical clarity

Expected Output


CentimetersToInchesConverter.java 1
NameFormatter.java 1
ai assistant > VowelCounter.java > ...
Run main | Debug main | Run | Debug
PROBLEMS 7
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Run:

```java
//• Use zero-shot prompting to generate a function that
//• Use few-shot prompting for the same problem
//• Compare both outputs based on:
//o Accuracy
//o Readability
//o Logical clarity

import java.util.Scanner;

public class VowelCounter {
    public static int countVowels(String str) {
        int count = 0;
        String vowels = "aeiouAEIOU";
        for (char c : str.toCharArray()) {
            if (vowels.indexOf(c) != -1) {
                count++;
            }
        }
        return count;
    }

    public static void main(String[] args) {
```

```
+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\WINDOWS\AppD
aceStorage\1e0d68f7cdf4b80abc760bf2e27fed0f\redhat.java\jdt_ws\de
unter'
Enter a string: ai assistant
Number of vowels in the string: 5
PS C:\dev ops> 9
```

2.vowel count using few shot

Task 5: Few-Shot Prompting – File Handling

Scenario

File processing requires clear logical understanding.

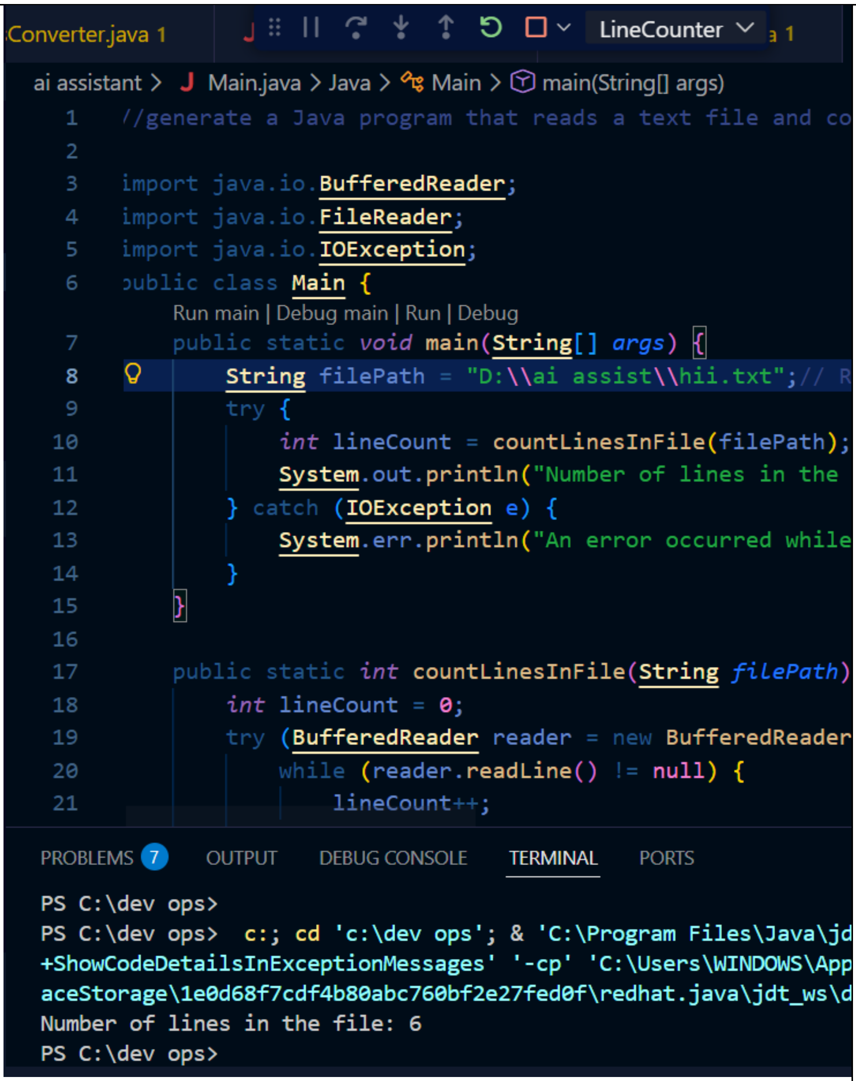
Task Description

Use few-shot prompting to generate a Python function that:

- Reads a .txt file
- Counts the number of lines in the file
- Returns the line count

Expected Output

Prompt: //generate a Java program that reads a text file and counts the number of lines in it. Counts the lines and prints the result to the console. code reads the file name as input from the user from the system and access the file to count the lines.


Converter.java 1
LineCounter
ai assistant > Main.java > Java > Main > main(String[] args)
Run main | Debug main | Run | Debug
PROBLEMS 7
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```java
//generate a Java program that reads a text file and co

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
public class Main {
    public static void main(String[] args) {
        String filePath = "D:\\ai assist\\hii.txt";// R
        try {
            int lineCount = countLinesInFile(filePath);
            System.out.println("Number of lines in the
        } catch (IOException e) {
            System.err.println("An error occurred while
        }
    }

    public static int countLinesInFile(String filePath)
        int lineCount = 0;
        try (BufferedReader reader = new BufferedReader
            while (reader.readLine() != null) {
                lineCount++;
```

```
PS C:\dev ops>
PS C:\dev ops>  c:; cd 'c:\dev ops'; & 'C:\Program Files\Java\jd
+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\WINDOWS\App
aceStorage\1e0d68f7cdf4b80abc760bf2e27fed0f\redhat.java\jdt_ws\d
Number of lines in the file: 6
PS C:\dev ops>
```

Note: Report should be submitted as a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots.