Task 8. Implemat various text file operation.

Aim: To write a Python program Implement various text file operations.

Problem 8.1:
you need to write the sentence "Error objects are thrown when runtime errors occur. The error object can also be used as a base object for user-defined exceptions" into a text file named log.txt. Implement a function that performs this task.

Algorithm:

1. Write to a File:
   - Define writefile(filename) function:
     - Open a file named "log.txt" in write mode.
     - write the following text to the file:
       - Error objects are thrown when runtime errors occurs. the Error object can also be used as a base object for user defined exceptions"
     - close the file.

2. Read from a file:
   - Define readfile(filename) function:
     - Open the file specified by filename in read mode using a with statement.
     - Read the entire content of the file
     - Print the content

3. Execute the Program:
   - call writefile("write") to write the predefined text to "log.txt".
   - call readfile("text") to attempt to read from a file named "text" and print its content.

Output:

Error objects are thrown when runtime errors occur. The Error objects can also be used as a base for user-defined exceptions.

Program: 8.1

```
def write_file(filename):
    f = open("log.txt","w")
    f.write("Errors Object are thrown when runtime error occur. The
      Error object can also be used as a base object for
      user-defined exceptions")
    f.close()
def readfile(filename):
    with open(filename,"r") as file:
    Content = file.read()
    Print(Content)
writefile("write")
readfile("text")
```

Problem 6.2:
you have a text.file log.txt containing logs of a system. write a function that counts the number of lines containing the word "ERROR".

Algorithm:

1. Initialize Error Counter:
   - Define the function count_error_lines(filename):
     - Initialize error_count to 0.
2. open and Read File:
   - Open the file specified by filename in read mode using a with statement.
3. Check each line for "Error":
   - Loop through each line in the file:
     - If the line contains the word "ERROR", increment error_count by 1.
4. Return Error Count:
   - After reading all the lines, return the value of error_count

output:

Number of lines with "ERROR" is 2

5. Execute the program:

- Call count_error_lines("log.txt") to count the number of lines with the word "Error", increment error_count by 1.

Program 8.3:

```
def count_error_lines(filename):
    error_count = 0
    with open(filename, "r") as file:
        for line in file:
            if "ERROR" in line:
                error_count += 1
    return error_count
error_lines = count_error_lines("log.txt")
print(f"Number of lines with 'ERROR': {error_lines}")
```

log.txt

"Error objects are thrown when runtime Error occur. The Error Objects can also be used as a base object for user defined exceptions

Problem 8.3

You need to write a report containing the details (name, departments) of the employee in list. Write a python function that writes this report to a file named.

employee_report.txt

Algorithm:

1. Create Employee Data:
    - Define the function write_employee_report(file name):
        - Create a list employees containing dictionaries, each with "name" and "department" keys for individual employees.
2. Open File for writing:
    - Open the file specified by filename in write mode using a with Statement.

Output:

Name: Alice, Department: HR

Name: Bob, Department: Engineering

Name: Charlie, Department: Finance

3. Write Employee data to file:

- Loop through each employee in the employee list:
  - For each employee, format a string as "Name: {employee['name']}, Department: {employee['department']}".
  - Write the formatted string to the file followed by a newline character (\n).

4. Execute the Program
   - Call write_employee_report("employee_report.txt") to write the employee data to the file "employee_report.txt".

Program 8.3:

```
def write_employee_report(filename):
    employees = [
        {"name": "Alice", "department": "HR"},
        {"name": "Bob", "department": "Engineering"},
        {"name": "Charlie", "department": "Finance"}
    ]
    with open(filename, "w") as file:
        for employee in employees:
            line = f"Name: {employee['name']}, Department: {employee['department']}\n"
            file.write(line)
```

VEL TECH - CSE	
EX NO.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	15
SIGN WITH DATE	

Result: Thus, the Python Program to implement various textfile operations was successfully executed and the output was verified.