Task : 6 Date: 01/09/25

Task 6 - Implement various text file operations.

Problem 6.1: 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.

Aim:- To write a python program implement various text file operations.

Algorithm:-

1. Write to a file.
   - Define write file (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 occur. The error object can also be used as a base object for user-defined expections"
   - close the file.
2. Read from a file.
   - Define readfile (filename) functions:
   - 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.

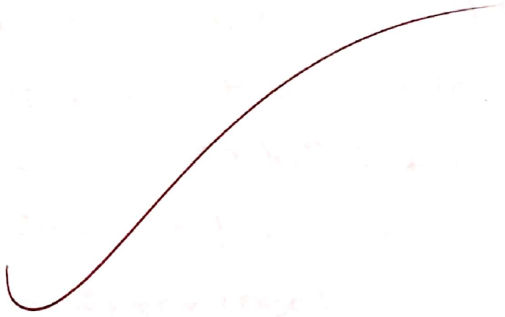
Program:

```
def writefile (filename):
    f = open ("log.txt","w")
    f.write ("Error objects are thrown when runtime
    errors occur. The error object can also be used as
    a base object for user-defined expections")
```

output:-

Error objects 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).
write file("write").
read file("text").
```



Result:- Thus, the Python program implement various text file operations was successfully executed and the output was verified.

VEL TECH	
EX.No.	
PERFORMANCE (5)	
RESULT AND ANALYSIS (5)	
VIVA VOCE (5)	
RECORD (5)	
TOTAL (30)	
SIGN WITH DATE	

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".

Aim: To write a python program implement various text file operations.

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.

5. Execute the program.

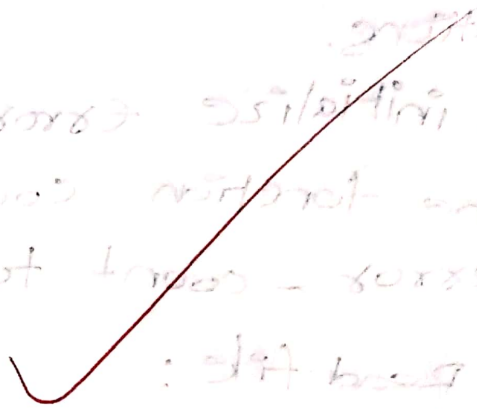
- Call count_error_lines("log.txt") to count the number of lines with the word "ERROR in the file "log.txt".
- Print the result with the message: "Number of lines with 'ERROR': {error_lines}".

Program:-

```
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
```

Output:-

Number of lines with 'ERROR' is 2.

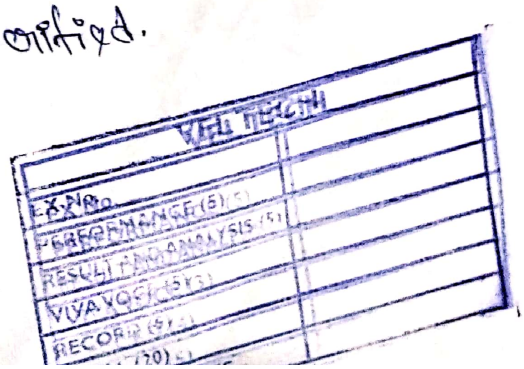


```
error_line:
count_error_line("log.txt").
print(f"Number of lines with 'ERROR':
  {error_line}").
log.txt.
```

"ERROR objects are thrown when runtime error occur.

The error object can also be used as a base object for user-defined exceptions."

Result:- Thus, the python program implement various text file operations was successfully executed and the output was verified.



Problem 6.3 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.

Aim:- To write a python program implementing various text file operations.

Algorithm:-

1. create employee data:
   - Define the function write_employee_report. (filename):
   - 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.
3. write employee data to file:
   - loop through each employee in the employees 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".

Output:-

Name : Alice, Department : HR.

Name : Bob, Department : Engineering.

Name : Charlie, Department : finance.

08/9/25
in python

9.

Program.

```
def write_employee_report(filen
    employees = [
{ "name": "Alice", "department":
{ "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).

# example usage:
write_employee_report("employee_report.txt")
```

VEL TECH	
EX No.	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	3
TOTAL (15)	13
SIGN WITH DATE	13/9/25

Result:- Thus, the Python program Implement various .txt file operations was successfully executed and the output was verified.

VEL TECH	
EX No.	
PERFORMANCE (5)	
RESULT AND ANALYSIS (5)	
VIVA VOCE (5)	
RECORD (5)	
TOTAL (20)	
SIGN WITH DATE	