* Program 6.1:-

```
def writefile(filename):
    f=open("log.txt","w").
    f.write("Error objects are thrown when runtime
    error occur. the error object can also be
    used as a base object for user-defined excep-
    -tions").
    f.close()
def readfile(filename):
    with open(filename,"r") as file:
    content=file.read()
    print(content)
    writefile("write")
    readfile("text").
```

- Output:-

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

Date:- 1/9/25

TASK-6:- Implement various text file operation

Aim:- To write a python programming to implem-ent various text file operations.

Problem:- 6.1:-

You need to write the sentence "Error objects are thrown when runtime errors occur. The ErrorObject Can also be used as a base object for user-defined exceptions.

Algorithm:-

1. write to a File:-
   - Define writefile (filename)-function:
     - open a file name "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 exceptions".
     - Close the file
2. Read form a file:-
   - Define readfile (filenam) function:
     - open the file specified by filename in read mode using a with statement.
     - Read the entire Content of 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 6.3:-

```
def Count_error_lines (filename):
    error_count = 0
    with open (filename, "r") as file:
        for line in life:
            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 object can also be used as a base object for user defined exceptions."

output:-

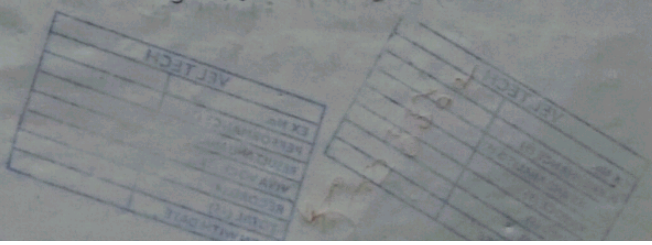
Number of lines with Error is 2

Program 6.2

You have a text file log.txt containing logs of a system. write a function that counts the number of lines containing 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 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_eerror_Lines ("log.txt") to Count the number of lines with the word "ERROR" in the file "log.txt".



*Program:-

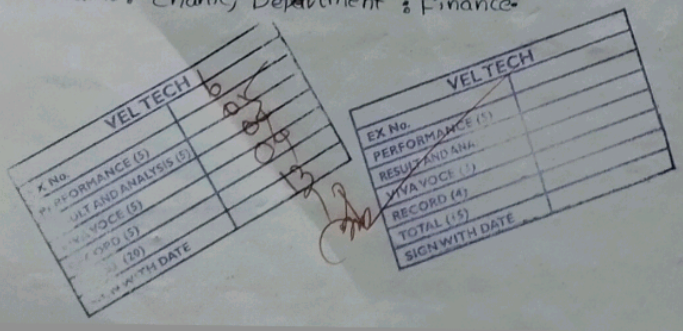
```
def write_employee_report (filename):
    employees = [{"name": "Alice", "department": "HR"},
    {"name": "Bob", "department": "Engineering"},
    {"name": "Charlie", "department": "Finance"}].

    with open (filename, "w") as file :
        for employee in employees:
        line = f"Name: {employee['name']}, Department :
            {employee['department']} \n".
        file.write (line).

#Example usage:-
write_employee_report ("employee_report.txt").
```

Output:-

Name : Alice, Department : HR

Name : Bob, Department : Engineering

Name : Charlie, Department : Finance.



Program 6.3:-

You need to write a report containing the details 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 (filename)
     - Create a list employees containing dictionaries each with "name" and "department" key for individual employees.
2. Open file for working :
   - 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 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".

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