Task 8 :- Implement various text file operation.

15/9/25 (8.1).

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

Algorithm:

1. write to file:
   - Define write file (filename) function;
   - open a file named "log.txt" in write mode.
   - write the following text to file.
     - Error objects are thrown then runtime errors occur. The object can also be base object for user-defined
   - 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 file.
   - print content.
3. Execute program:
   - call writefile (write) to write predefined text to log.txt.
   - call readfile ("text") to attempt to read a file named "text" and print its content.
   - ~~call readfile~~ (

program 8.1

```
def writefile (filename).
   f= open ("log.txt, "w")
   f. write ("Error objets are thrown when runtime error
occur. The error object can also be used base object user-defined
exceptions").
   f. close().
def- readfile (file name):
      with  open (filename, "r") as file:
```

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

outputs

Number of lines with 'ERROR' is 2.

print (content)

write file ("rewrite")
read file ("text")

problem_8.2:-

You have 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 using a with statement.
3. check each line for "error":
    - loop through each line in file.
        - If the line contains the word "ERROR." Increment error-count by 1.
4. Return Error count:
    - After reading all lines, return value of error-count
5. Execute program:
    - call count-error-lines ("log.txt") to count the number of line with the word "Error" in file "log.txt".
    - print result with message. "Number of lines with Error"

Program:-

```
def count-error-lines(filename):
    error-count =0
    with open(file name,"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}")
```

output:

```
Name: Alice, Department: HR.
Name: Bob, Department: Engineering
Name: charlie, Department: Finace.
```

log. txt

"Error" objects are thrown when runtime Error occur

the Error object can be used as based object for use-defined exceptions".

Problem 8.3:-

you need to write report containing the details of employee in list. write python that write this report to file named Employee - report.txt.

~~Alg~~ Algorithm:

1. Create Employee Data:
   - Define the function write employee_ report (filename);
   - Create a list employee containing dictionaries each with "name" and "department" keys for individual Empolyees.
2. open file for writing:
   open the file specified by filename write_employee_report.
3. ● write employee Data to file:
   - loop through each Employee in employee list:
   - For each employee, format a string as "Name":{ employee ('Name')}.
   - write the formatted string to file, followed by a newline character (\n).
4. Execute the program:
   - call write_Employee_report ("employee-report.txt") to write employee data to file "employee - report.txt".

program:

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

```
file.write(Name).

# example usage.
write_employee_report("employee.txt").
```

VEL TECH - CSE	
EX NO.	
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 implement various text file operation was successfully and output was verified.