Modern Education Society's Wadia College of Engineering, Pune.

Department of Computer Engineering

NAME OF STUDENT	CLASS
SEMESTER/YEAR	ROLL NO
DATE OF PERFORMANCE	DATE OF SUBMISSION
EXAMINED BY	EXPERIMENT NO

Assignment No-4

Title: Write a map-reduce program to count the number of occurrences of each alphabetic character in the given dataset. The count for each letter should be case-insensitive (i.e., include both upper- case and lower-case versions of the letter; Ignore non-alphabetic characters).

Objectives:

- To learn how to use Map-Reduce for text processing.
- To count the number of occurrences of each alphabetic character in a given dataset in a case-insensitive manner.
- To understand the concept of ignoring non-alphabetic characters during text analysis.

Outcomes:

- Understand the Map-Reduce programming paradigm.
- Implement a program to process textual data.
- Analyze frequency distribution of letters in text.
- Apply case-insensitive operations in text processing.

Tools Required:

- **Software:**
  - Open-source operating system (Linux/Windows/Mac)
  - Python 3.x
  - Hadoop (optional for full Map-Reduce environment)
  - Python libraries: `mrjob` (for local Map-Reduce simulation), or plain Python for simplified implementation

Theory:

Map-Reduce:
Map-Reduce is a programming model used for processing large datasets in parallel. It consists of two main steps:

1. **Map:** Processes input data and produces key-value pairs.
2. **Reduce:** Aggregates the key-value pairs to produce final output.

In this assignment, each alphabetic character is treated as a key, and the count is the value. Map-Reduce will help efficiently compute counts for large datasets.

Example:
Suppose the dataset is:

Hello World!

After converting to lowercase and ignoring non-alphabetic characters, we get:

h e l l o w o r l d

The frequency of each character:

d:1, e:1, h:1, l:3, o:2, r:1, w:1

Python Map-Reduce Program:
Here is a sample Python program using `mrjob` library:

```
from mrjob.job import MRJob

import re

WORD_RE = re.compile(r'[a-zA-Z]')  # Only alphabetic characters

class MRCharCount(MRJob):

    def mapper(self, _, line):

        for char in line:

            if WORD_RE.match(char):

                yield (char.lower(), 1)

    def reducer(self, key, values):

        yield (key, sum(values))

if __name__ == '__main__':

    MRCharCount.run()
```

Conclusion:

After completing this experiment, we have learned:

- How to implement a simple Map-Reduce program in Python.
- How to count alphabetic characters case-insensitively.
- How to ignore non-alphabetic characters in text processing.

Questions:

Q.1) How does the Map-Reduce framework help in processing large datasets?

Q.2) Modify the program to count the frequency of words instead of characters.

Q.3) Explain how you can extend this program to ignore punctuation and numbers while counting words.

Q.4) Why is it important to make the character count case-insensitive?