```
from collections import defaultdict

# Mapper function
def mapper(line):
    line = line.lower()

    for char in line:
        if char.isalpha():
            yield (char, 1)

# Reducer function
def reducer(key, values):
    return (key, sum(values))

# MapReduce simulation
def map_reduce(input_data):
    intermediate = defaultdict(list)
    for line in input_data:
        for key, value in mapper(line):
            intermediate[key].append(value)

    results = []
    for key in intermediate:
        results.append(reducer(key, intermediate[key]))

    return results

# Function to read data from a text file and process it
def process_file(file_path):
    with open(file_path, 'r') as file:
        lines = file.readlines()

    output = map_reduce(lines)

    for char, count in output:
        print(f"({char}: {count})")

file_path ='/home/kunalpisolkar/Documents/Practicals/IR_Lab/four/document1.txt'
process_file(file_path)
```

Output:

```
(a: 27)
(r: 17)
(t: 24)
(i: 32)
(f: 6)
(c: 12)
(l: 12)
(n: 26)
(e: 29)
(g: 13)
(d: 6)
(m: 8)
(h: 11)
(s: 22)
(o: 16)
(u: 12)
(b: 4)
(p: 5)
(k: 2)
(w: 2)
(x: 1)
(v: 3)
(y: 1)
(z: 1)
```