**EXPERIMENT 5 MAPREDUCE**

from collections import defaultdict

from multiprocessing import Pool

def mapper(text):

words = text.split()

word\_count = defaultdict(int)

for word in words:

word\_count[word] += 1

return word\_count

def reducer(word\_counts):

total\_word\_count = defaultdict(int)

for word\_count in word\_counts:

for word, count in word\_count.items():

total\_word\_count[word] += count

return total\_word\_count

if \_\_name\_\_ == '\_\_main\_\_':

num\_inputs = int(input("Enter the number of input strings: "))

inputs = []

for i in range(num\_inputs):

input\_text = input(f"Enter input string {i + 1}: ")

inputs.append(input\_text)

with Pool(processes=len(inputs)) as pool:

mapped\_results = pool.map(mapper, inputs)

reduced\_result = reducer(mapped\_results)

for word, count in reduced\_result.items():

print(f"{word}: {count}")