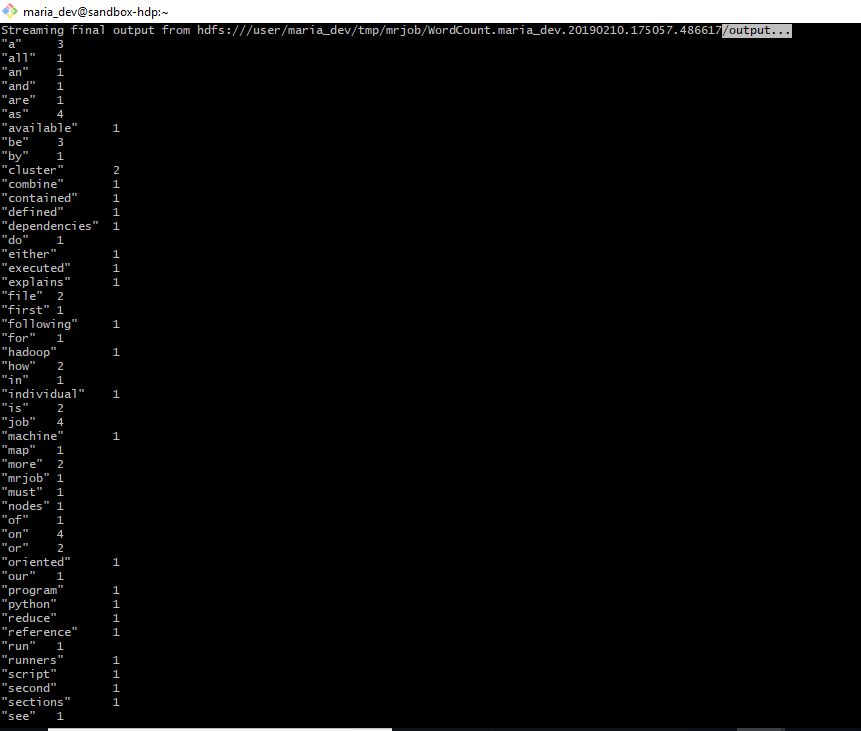
Q1. Submit a copy of this modified program and a screen shot of the results of the program’s execution as the output of your assignment.

**Output for WordCount.py**



**Updated Code: WordCount2.py**

from mrjob.job import MRJob

import re

WORD\_RE = re.compile(r"[\w']+")

class MRWordCount(MRJob):

def mapper(self, \_, line):

for word in WORD\_RE.findall(line):

if(word[0]>="a" and word[0]<="n"):

yield "a-n",1

else:

yield "other",1

def combiner(self, word, counts):

yield word, sum(counts)

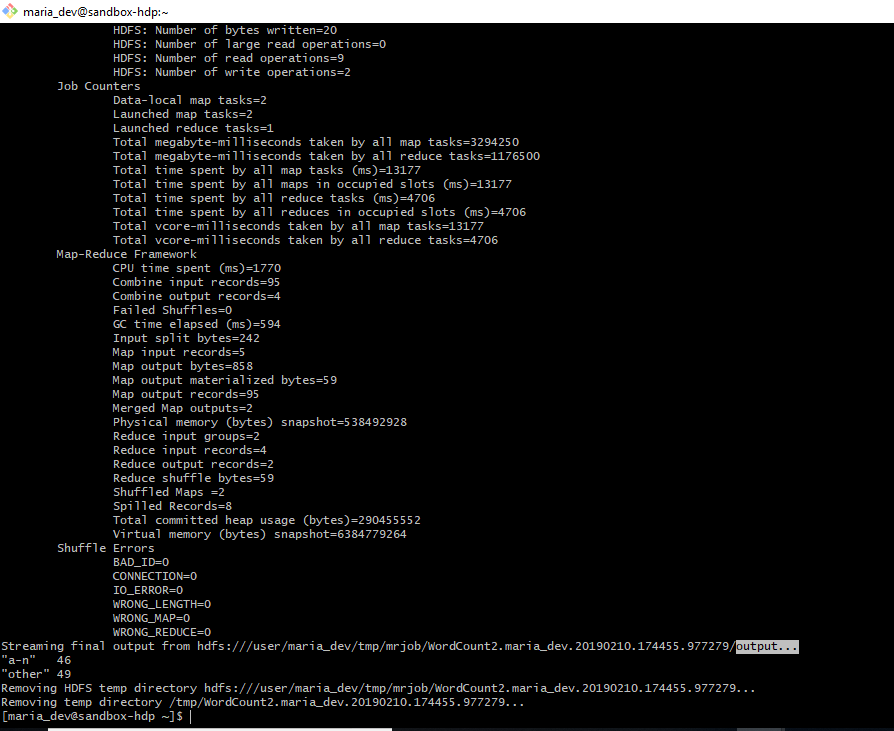
def reducer(self, word, counts):

yield word, sum(counts)

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

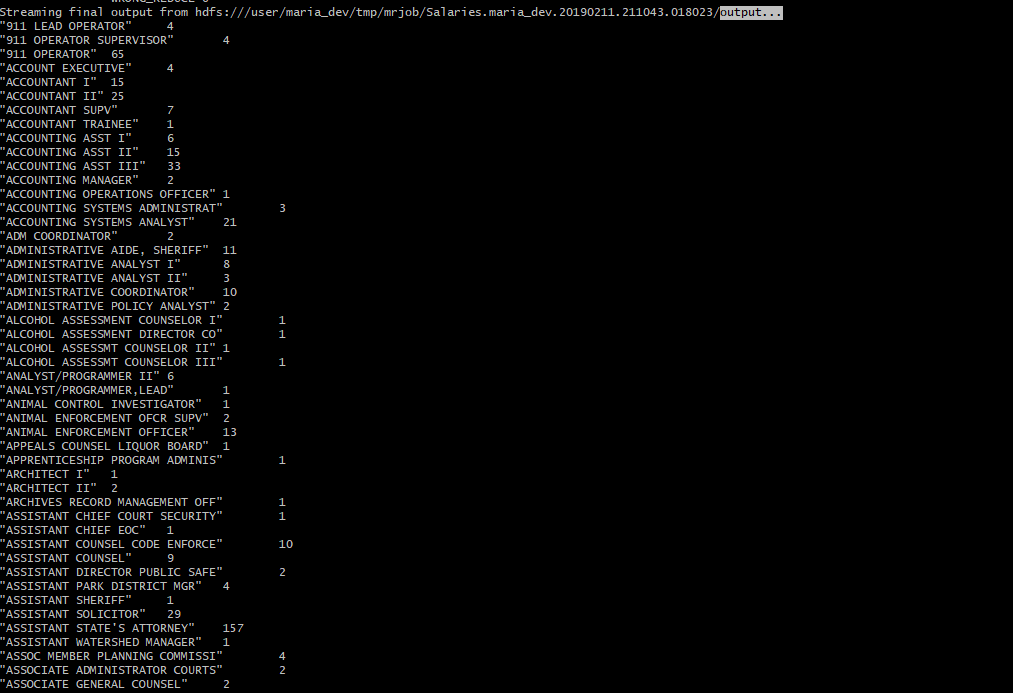
MRWordCount.run()

**WordCount2.py**

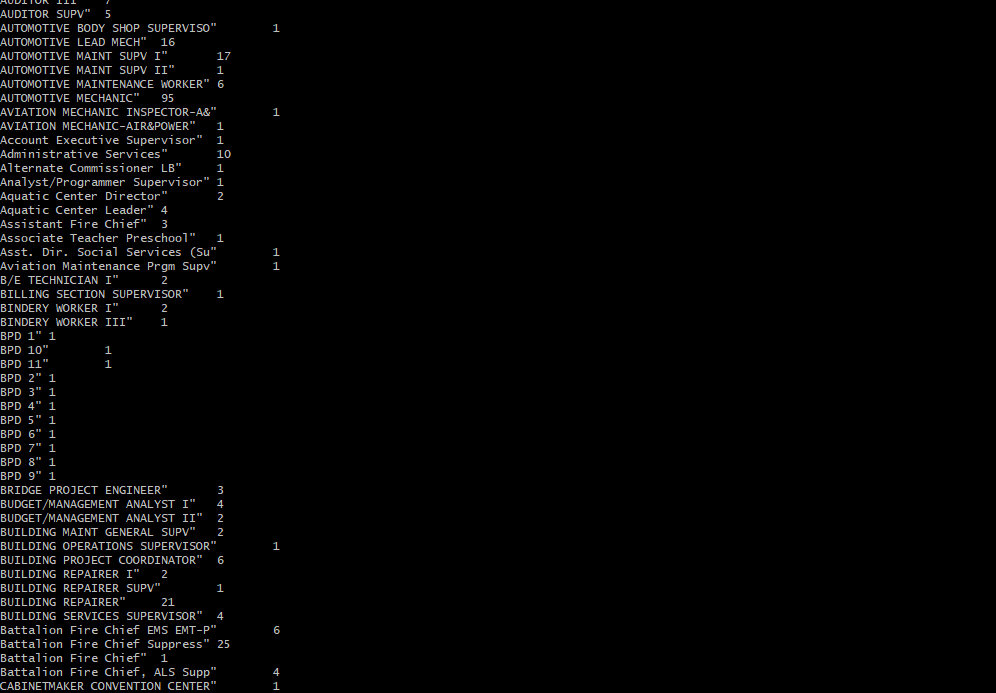


Q2. Submit a copy of this modified program and a screen shot of the results of the program’s execution as the output of your assignment.

**Salaries.py**



Salaries.py (Rest output)



**Updated Code: Salaries2.py**

from mrjob.job import MRJob

class MRSalaries(MRJob):

def mapper(self, \_, line):

(name,jobTitle,agencyID,agency,hireDate,annualSalary,grossPay) = line.split('\t')

if (float(annualSalary) >= 0.00 and float(annualSalary) <= 49999.99):

yield "low", 1

elif (float(annualSalary) >= 50000.00 and float(annualSalary) <= 99999.99):

yield "medium", 1

elif (float(annualSalary) >= 100000.00):

yield "high", 1

def combiner(self, salary, counts):

yield salary, sum(counts)

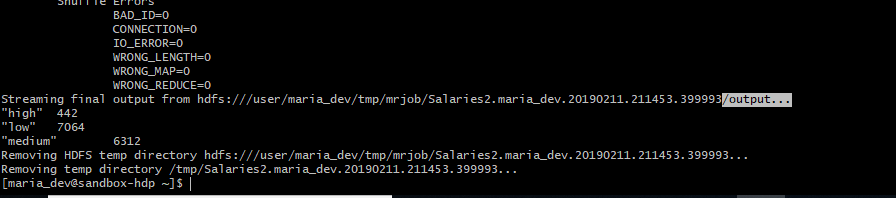
def reducer(self, salary, counts):

yield salary, sum(counts)

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

MRSalaries.run()

**Output for Slalaries2.py**



**Q3.** Review the slides 17-22 in lecture notes Module 3b. Now write a program to perform the task of outputting a count of the number of movies each user (identified via their user id) reviewed.

from mrjob.job import MRJob

from mrjob.step import MRStep

class Ratings(MRJob):

def mapper(self, \_, line):

(userID, movieID, rating, timestamp) = line.split(',')

yield int(userID), 1

def combiner(self, userID, counts):

yield int(userID), sum(counts)

def reducer(self, userID, counts):

yield int(userID), sum(counts)

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

Ratings.run()

**Ratings.py**



