Ayush Goyal

190905522 CSE D 62

Distributed Systems Week 5: Lab 5: Map Reduce Programming using Python

Mapper: A block of data is read and processed to produce key-value pairs as intermediate output. The output of mapper is given as input to reducer.

Reducer: Receives the key-value pair from multiple mappers. Then, the reducer aggregates those intermediate data tuples (intermediate key-value pair) into a smaller set of tuples or key-value pairs which is the final output.

1. Write a basic wordcount program.

Note: I have considered the 'age' column of the 'heart_disease_data.csv' file for this problem.

1mapper.py

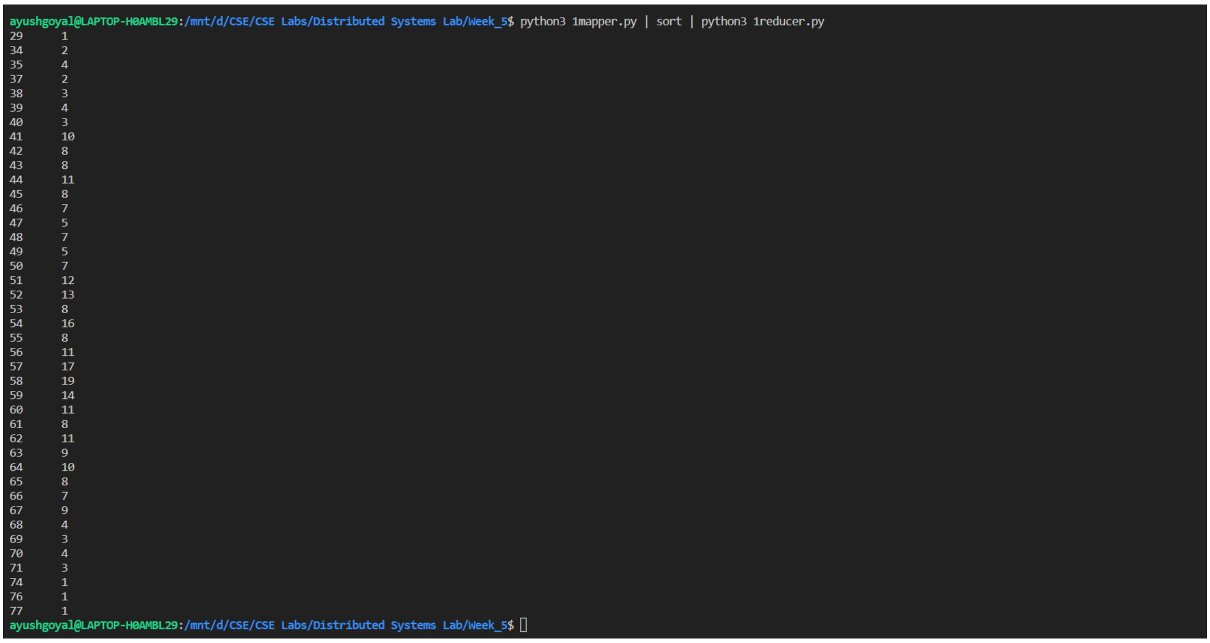
```
"""mapper.py"""
import sys
import pandas as pd
df = pd.read_csv('heart_disease_data.csv')
for age in df['age']:
        print('%s\t%s' %(age, 1))
```

1reducer.py:

```
"""reducer.py"""
import sys

current_word = None
current_count = 0
word = None

for line in sys.stdin:
        line = line.strip()
        word, count = line.split()
        try:
                count = int(count)
        except ValueError:
                continue

        if current_word == word:
                current_count += count
        else:
                if current_word:
                        print(current_word + '\t' + str(current_count))
                current_count = count
                current_word = word

print(current_word + '\t' + str(current_count))
```

Output:

command: python3 1mapper.py | sort | python3 1reducer.py



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ python3 1mapper.py | sort | python3 1reducer.py
29      1
34      2
35      4
37      2
38      3
39      4
40      3
41      10
42      8
43      8
44      11
45      8
46      7
47      5
48      7
49      5
50      7
51      12
52      13
53      8
54      16
55      8
56      11
57      17
58      19
59      14
60      11
61      8
62      11
63      9
64      10
65      8
66      7
67      9
68      4
69      3
70      4
71      3
74      1
76      1
77      1
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ 
```

2. MapReduce program to find frequent words.

Note: I have considered the ‘Country/Region’ column of the ‘covid_19_data.csv’ file for this problem.

2freqmap1.py

```
import pandas as pd
df = pd.read_csv('covid_19_data.csv')
for country in df['Country/Region']:
        country = country.strip("(),'")
        print('%s\t%d' %(country, 1))
```

2freqred1.py

```
import sys
lastWord = None
sum = 0
for line in sys.stdin:
        word, count = line.strip().split('\t', 1)
        count = int(count)

        if lastWord == None:
                lastWord = word
                sum = count
                continue

        if word == lastWord:
                sum += count
        else:
                print('%s\t%d' %(lastWord, sum))
                sum = count
        lastWord = word

if lastWord == word:
    print('%s\t%s' %(lastWord, sum))
```

2freqmap2.py

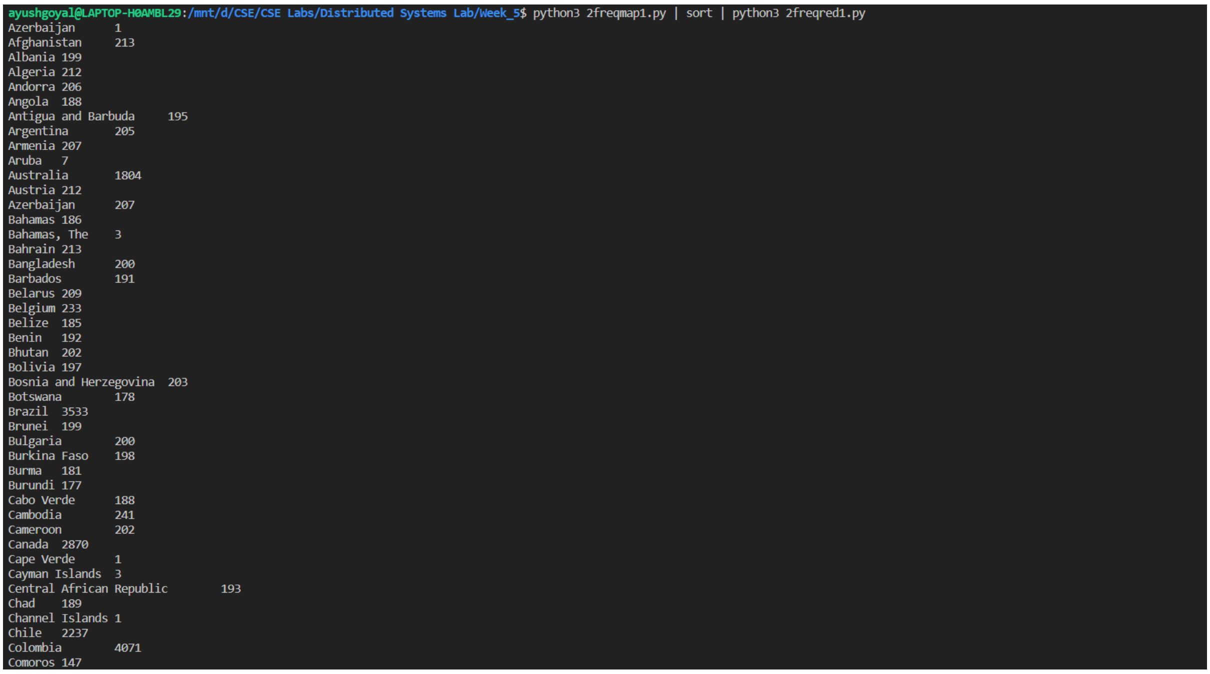
```
import sys
for line in sys.stdin:
    word, count = line.strip().split('\t', 1)
    count = int(count)
    print('%d\t%s' %(count, word))
```

2freqred2.py

```
import sys
mostFreq = []
currentMax = -1

for line in sys.stdin:
    count, word = line.strip().split('\t', 1)
    count = int(count)
    if count > currentMax:
        currentMax = count
        mostFreq = [word]
    elif count == currentMax:
        mostFreq.append(word)

for word in mostFreq:
    print('%s\t%s' %(word, currentMax))
```

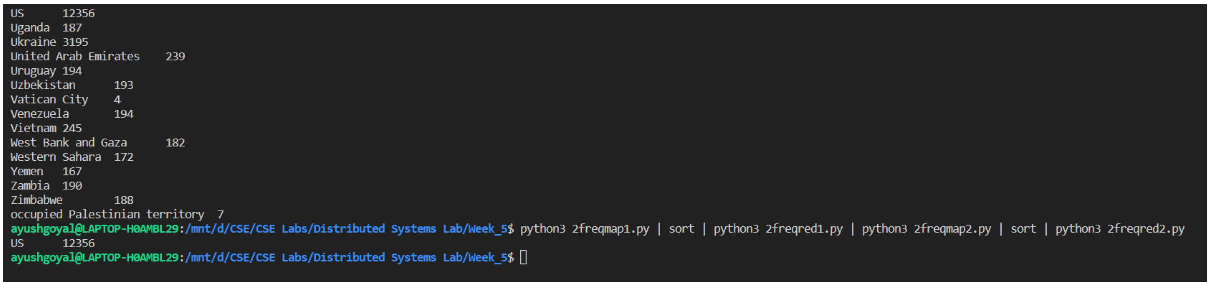
Output:

command: python3 2freqmap1.py | sort | python3 2freqred1.py



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ python3 2freqmap1.py | sort | python3 2freqred1.py
Azerbaijan      1
Afghanistan     213
Albania 199
Algeria 212
Andorra 206
Angola  188
Antigua and Barbuda     195
Argentina       205
Armenia 207
Aruba   7
Australia       1804
Austria 212
Azerbaijan      207
Bahamas 186
Bahamas, The    3
Bahrain 213
Bangladesh      200
Barbados        191
Belarus 209
Belgium 233
Belize  185
Benin   192
Bhutan  202
Bolivia 197
Bosnia and Herzegovina  203
Botswana        178
Brazil  3533
Brunei  199
Bulgaria        200
Burkina Faso    198
Burma   181
Burundi 177
Cabo Verde      188
Cambodia        241
Cameroon        202
Canada  2870
Cape Verde      1
Cayman Islands  3
Central African Republic        193
Chad    189
Channel Islands 1
Chile   2237
Colombia        4071
Comoros 147
```

command: python3 2freqmap1.py | sort | python3 2freqred1.py | python3 2freqmap2.py | sort | python3 2freqred2.py



```
US      12356
Uganda  187
Ukraine 3195
United Arab Emirates    239
Uruguay 194
Uzbekistan      193
Vatican City    4
Venezuela       194
Vietnam 245
West Bank and Gaza      182
Western Sahara  172
Yemen   167
Zambia  190
Zimbabwe        188
occupied Palestinian territory  7
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ python3 2freqmap1.py | sort | python3 2freqred1.py | python3 2freqmap2.py | sort | python3 2freqred2.py
US      12356
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ 
```

3. Map Reduce program to explore the dataset and perform the filtering (typically creating key/value pairs) by mapper and perform the count and summary operation on the instances.

Note: I have considered the 'cost' column of the 'example.txt' file for this problem.

3itemmap.py

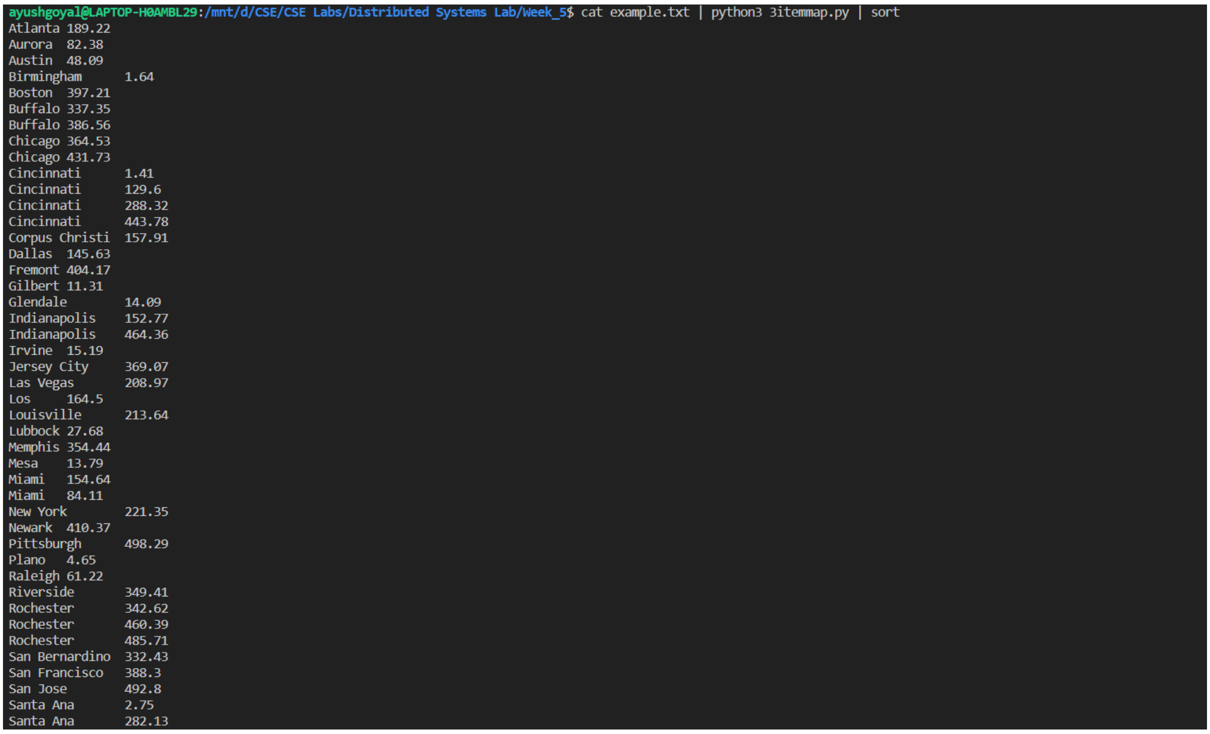
```
import fileinput
for line in fileinput.input():
  data = line.strip().split('\t')
  if len(data) == 6:
    date, time, location, item, cost, payment = data
    print("{0}\t{1}".format(location, cost))
```

3itemred.py

```
import fileinput
transactions_count = 0
sales_total = 0
for line in fileinput.input():
        data = line.strip().split('\t')
        if len(data) != 2:
                continue
        current_key, current_value = data
        transactions_count += 1
        sales_total += float(current_value)
print(transactions_count, '\t', sales_total)
```

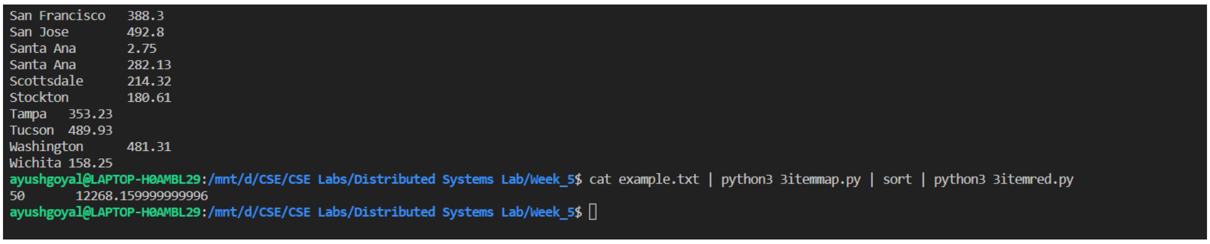
Output:

command: cat example.txt | python3 3itemmap.py | sort



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ cat example.txt | python3 3itemmap.py | sort
Atlanta 189.22
Aurora  82.38
Austin  48.09
Birmingham      1.64
Boston  397.21
Buffalo 337.35
Buffalo 386.56
Chicago 364.53
Chicago 431.73
Cincinnati      1.41
Cincinnati      129.6
Cincinnati      288.32
Cincinnati      443.78
Corpus Christi  157.91
Dallas  145.63
Fremont 404.17
Gilbert 11.31
Glendale        14.09
Indianapolis    152.77
Indianapolis    464.36
Irvine  15.19
Jersey City     369.07
Las Vegas       208.97
Los     164.5
Louisville      213.64
Lubbock 27.68
Memphis 354.44
Mesa    13.79
Miami   154.64
Miami   84.11
New York        221.35
Newark  410.37
Pittsburgh      498.29
Plano   4.65
Raleigh 61.22
Riverside       349.41
Rochester       342.62
Rochester       460.39
Rochester       485.71
San Bernardino  332.43
San Francisco   388.3
San Jose        492.8
Santa Ana       2.75
Santa Ana       282.13
```

command: cat example.txt | python3 3itemmap.py | sort | python3 3itemred.py



```
San Francisco   388.3
San Jose        492.8
Santa Ana       2.75
Santa Ana       282.13
Scottsdale      214.32
Stockton        180.61
Tampa   353.23
Tucson  489.93
Washington      481.31
Wichita 158.25
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ cat example.txt | python3 3itemmap.py | sort | python3 3itemred.py
50      12268.159999999996
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ ▯
```

4. Write a mapper and reducer program for word count by defining separator instead of using ‘\t’.

Note: I have considered the ‘DurationOfCreditInMonths’ column of the ‘German Credit.xlsx’ file for this problem.

<u>4sepmap.py</u>

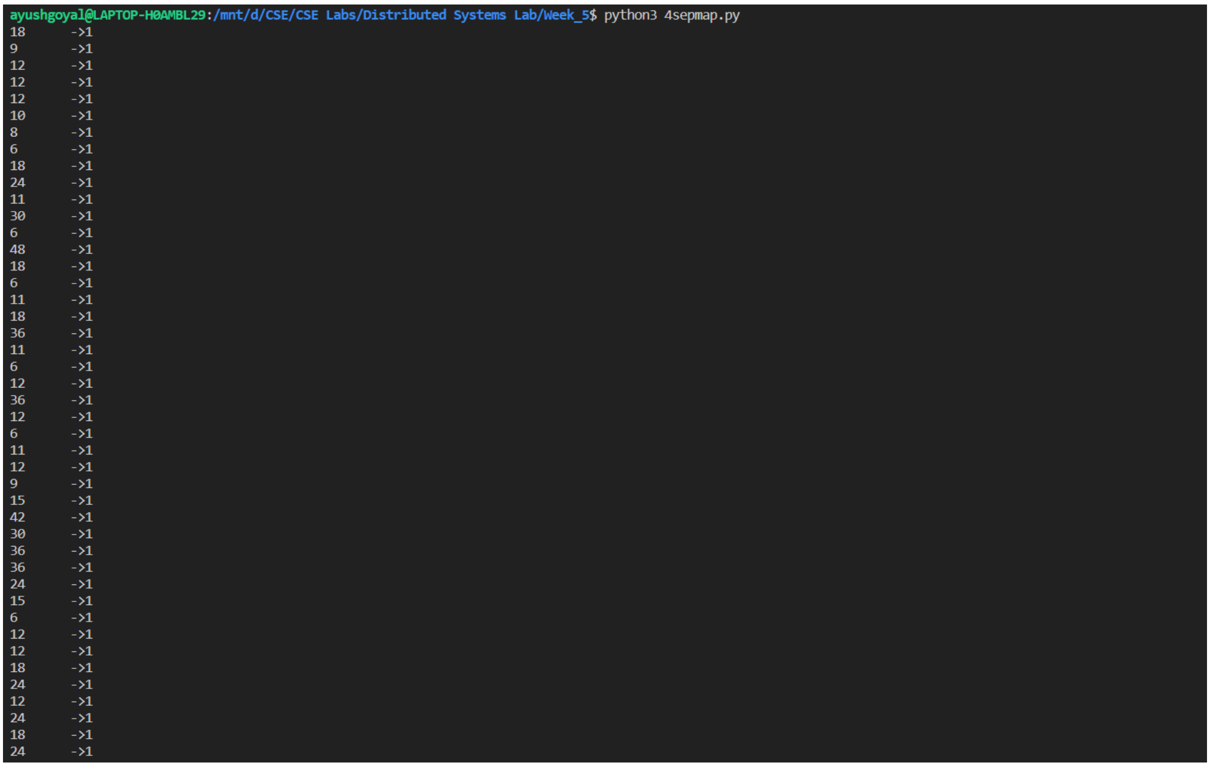
```
"""A more advanced mapper using python iterators and generators"""
import sys
import pandas as pd

def main(seperator = '\t'):
        G = pd.read_excel('German Credit.xlsx', sheet_name = 'Sheet1')
        for index, row in G.iterrows():
                print('%d%s%d' %(row['DurationOfCreditInMonths'], seperator, 1))

if __name__ == '__main__':
        main('\t->')
```

4sepred.py

```
from itertools import groupby
from operator import itemgetter
import sys

def red_mapper_output(file, seperator = '\t'):
    for line in file:
        yield line.rstrip().split(seperator, 1)

def main(seperator = '\t'):
    data = red_mapper_output(sys.stdin, seperator)
    for current_word, group in groupby(data, itemgetter(0)):
        try:
            total_count = sum(int(count) for current_word, count in group)
            print('%s%s%d' %(current_word, seperator, total_count))
        except ValueError:
            pass

if __name__ == '__main__':
    main('\t->')
```

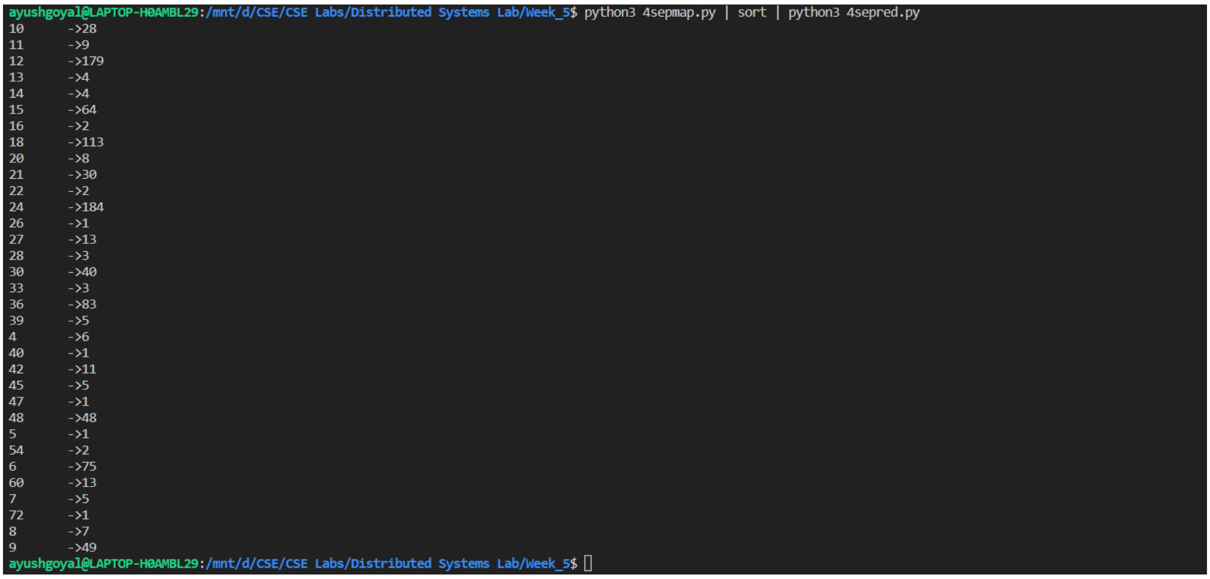
Output:

command: python3 4sepmap.py



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ python3 4sepmap.py
18      ->1
9       ->1
12      ->1
12      ->1
12      ->1
10      ->1
8       ->1
6       ->1
18      ->1
24      ->1
11      ->1
30      ->1
6       ->1
48      ->1
18      ->1
6       ->1
11      ->1
18      ->1
36      ->1
11      ->1
6       ->1
12      ->1
36      ->1
12      ->1
6       ->1
11      ->1
12      ->1
9       ->1
15      ->1
42      ->1
30      ->1
36      ->1
36      ->1
24      ->1
15      ->1
6       ->1
12      ->1
12      ->1
18      ->1
24      ->1
12      ->1
24      ->1
18      ->1
24      ->1
```

command: python3 4sepmap.py | sort | python3 4sepred.py



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ python3 4sepmap.py | sort | python3 4sepred.py
10      ->28
11      ->9
12      ->179
13      ->4
14      ->4
15      ->64
16      ->2
18      ->113
20      ->8
21      ->30
22      ->2
24      ->184
26      ->1
27      ->13
28      ->3
30      ->40
33      ->3
36      ->83
39      ->5
4       ->6
40      ->1
42      ->11
45      ->5
47      ->1
48      ->48
5       ->1
54      ->2
6       ->75
60      ->13
7       ->5
72      ->1
8       ->7
9       ->49
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ 
```

5. Write a map reduce program that returns the cost of the item that is most expensive, for each location in the dataset example.txt.

5itemmap_expensive.py

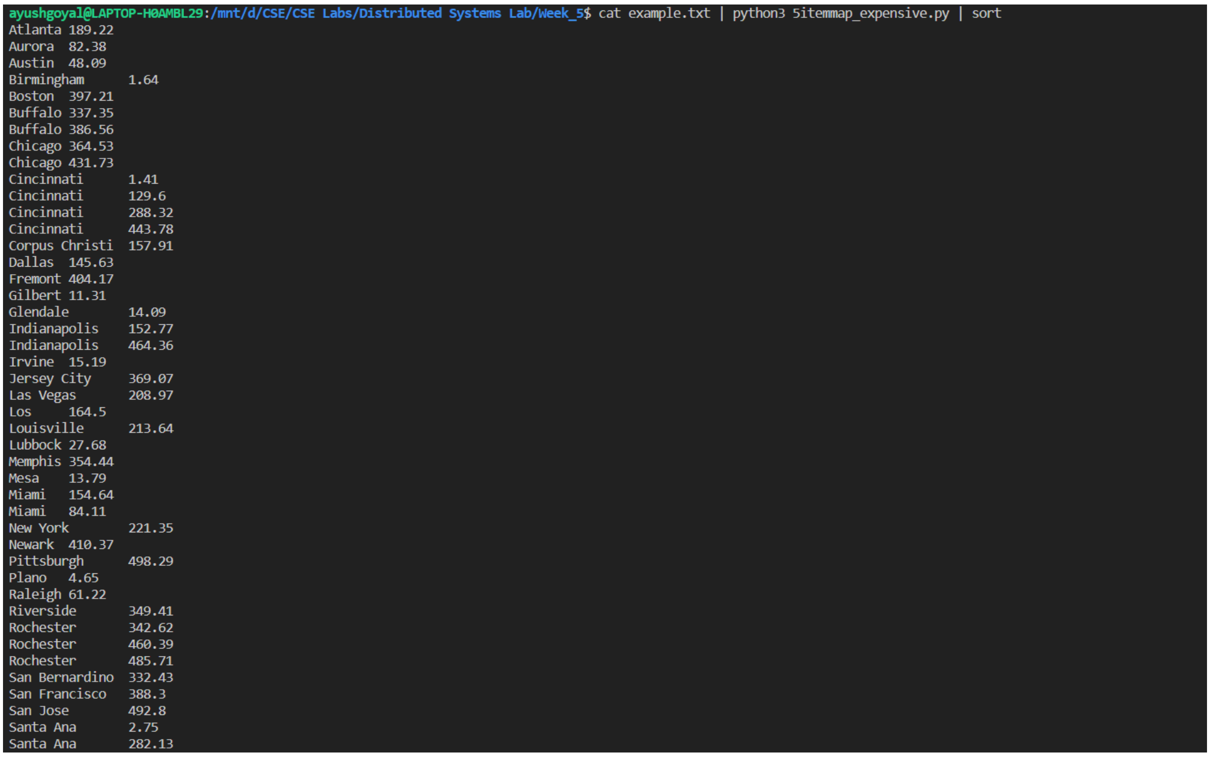
```
import fileinput
for line in fileinput.input():
        data = line.strip().split('\t')
        if len(data) == 6:
                date, time, location, item, cost, payment = data
                print('{0}\t{1}'.format(location, cost))
```

5itemred_expensive.py

```
import fileinput
max_value = 0
old_key = None

for line in fileinput.input():
        data = line.strip().split('\t')
        if len(data) != 2:
                continue

        current_key, current_value = data
        if old_key and old_key != current_key:
                print(old_key, '\t', max_value)
                old_key = current_key
                max_value = 0
        if float(current_value) > float(max_value):
                max_value = float(current_value)

if old_key != None:
        print(old_key, '\t', max_value)
```

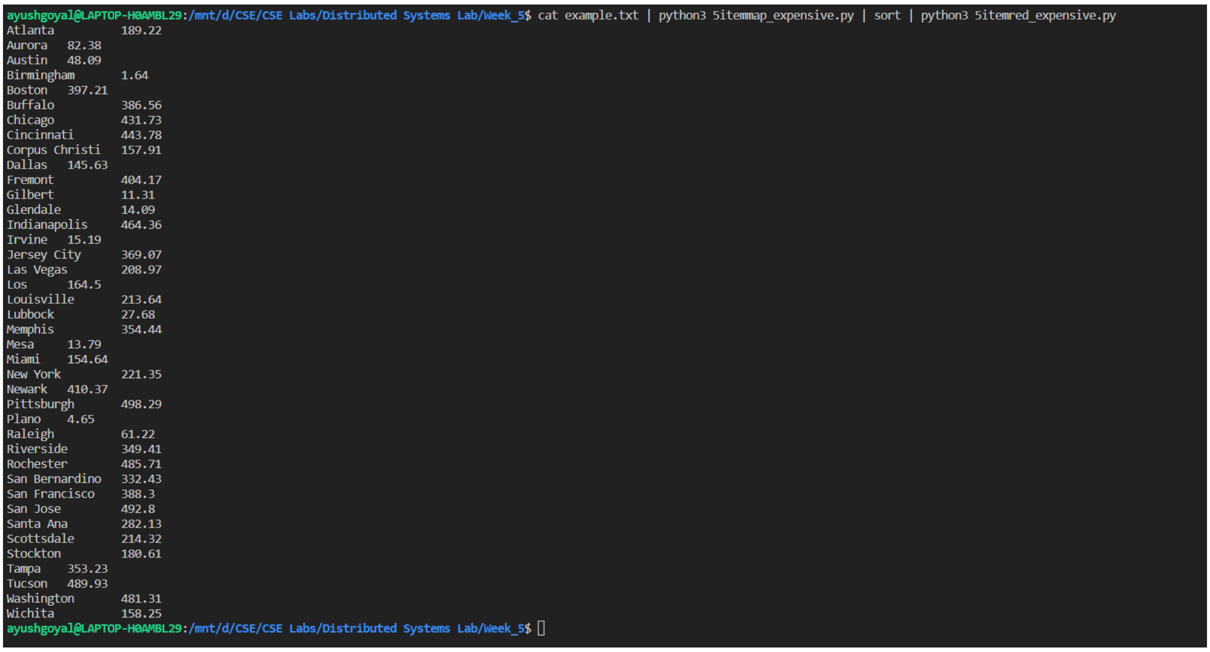
<u>Output:</u>

command: cat example.txt | python3 5itemmap_expensive.py | sort



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ cat example.txt | python3 5itemmap_expensive.py | sort
Atlanta 189.22
Aurora  82.38
Austin  48.09
Birmingham      1.64
Boston  397.21
Buffalo 337.35
Buffalo 386.56
Chicago 364.53
Chicago 431.73
Cincinnati      1.41
Cincinnati      129.6
Cincinnati      288.32
Cincinnati      443.78
Corpus Christi  157.91
Dallas  145.63
Fremont 404.17
Gilbert 11.31
Glendale        14.09
Indianapolis    152.77
Indianapolis    464.36
Irvine  15.19
Jersey City     369.07
Las Vegas       208.97
Los     164.5
Louisville      213.64
Lubbock 27.68
Memphis 354.44
Mesa    13.79
Miami   154.64
Miami   84.11
New York        221.35
Newark  410.37
Pittsburgh      498.29
Plano   4.65
Raleigh 61.22
Riverside       349.41
Rochester       342.62
Rochester       460.39
Rochester       485.71
San Bernardino  332.43
San Francisco   388.3
San Jose        492.8
Santa Ana       2.75
Santa Ana       282.13
```

command: cat example.txt | python3 5itemmap_expensive.py | sort | python3 5itemred_expensive.py



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ cat example.txt | python3 5itemmap_expensive.py | sort | python3 5itemred_expensive.py
Atlanta         189.22
Aurora  82.38
Austin  48.09
Birmingham      1.64
Boston  397.21
Buffalo         386.56
Chicago         431.73
Cincinnati      443.78
Corpus Christi  157.91
Dallas  145.63
Fremont         404.17
Gilbert         11.31
Glendale        14.09
Indianapolis    464.36
Irvine  15.19
Jersey City     369.07
Las Vegas       208.97
Los     164.5
Louisville      213.64
Lubbock         27.68
Memphis         354.44
Mesa    13.79
Miami   154.64
New York        221.35
Newark  410.37
Pittsburgh      498.29
Plano   4.65
Raleigh         61.22
Riverside       349.41
Rochester       485.71
San Bernardino  332.43
San Francisco   388.3
San Jose        492.8
Santa Ana       282.13
Scottsdale      214.32
Stockton        180.61
Tampa   353.23
Tucson  489.93
Washington      481.31
Wichita         158.25
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ 
```

6. Write a MapReduce program to evaluate value of PI.

6mapper_pi.py

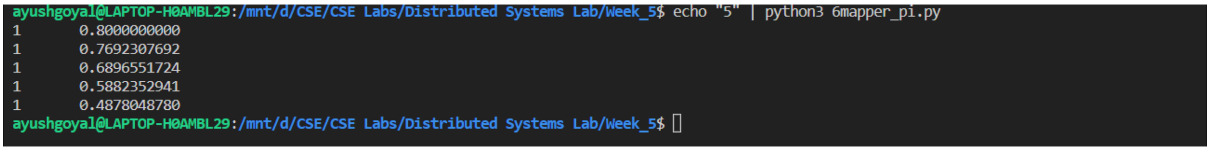
```
import sys
def f(x):
        return 4.0 / (1.0 + x * x)

for line in sys.stdin:
        words = line.strip().split()
        N = int(words[0])
        deltaX = 1.0 / N

        for i in range(N):
                print('1\t%1.10f' %(f(i * deltaX) * deltaX))
```

6reducer_pi.py

```
import sys
sum = 0
for line in sys.stdin:
        line = line.strip()
        word, count = line.split('\t', 1)
        try:
                count = float(count)
        except ValueError:
                continue
        sum += count

print('%1.5f\t0' %sum)
```

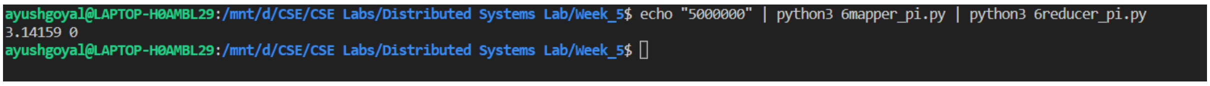
Output:

command: echo "5" | python3 6mapper_pi.py



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ echo "5" | python3 6mapper_pi.py
1       0.8000000000
1       0.7692307692
1       0.6896551724
1       0.5882352941
1       0.4878048780
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ 
```

command: echo "5000000" | python3 6mapper_pi.py | python3 6reducer_pi.py



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ echo "5000000" | python3 6mapper_pi.py | python3 6reducer_pi.py
3.14159 0
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ 
```

7. Write a MapReduce program to count even or odd numbers in randomly generated natural numbers.

8mapper1.py

```
"""mapper.py"""
import random
random.seed(0)
N = int(input())
for i in range(N):
        print(str(random.randrange(1, 10)), '\t', str(1))
```

8reducer1.py

```
"""reducer.py"""
import sys
lastNumber = 0
count = 0

for line in sys.stdin:
        curNumber, curCount = line.strip().split('\t')
        curNumber = int(curNumber)
        curCount = int(curCount)

        if count > 0 and lastNumber != curNumber:
                print('%d%s%d' %(lastNumber, '\t', count))
                count = 0
        lastNumber = curNumber
        count += curCount

if count > 0:
        print('%d%s%d' %(lastNumber, '\t', count))
```

8mapper2.py

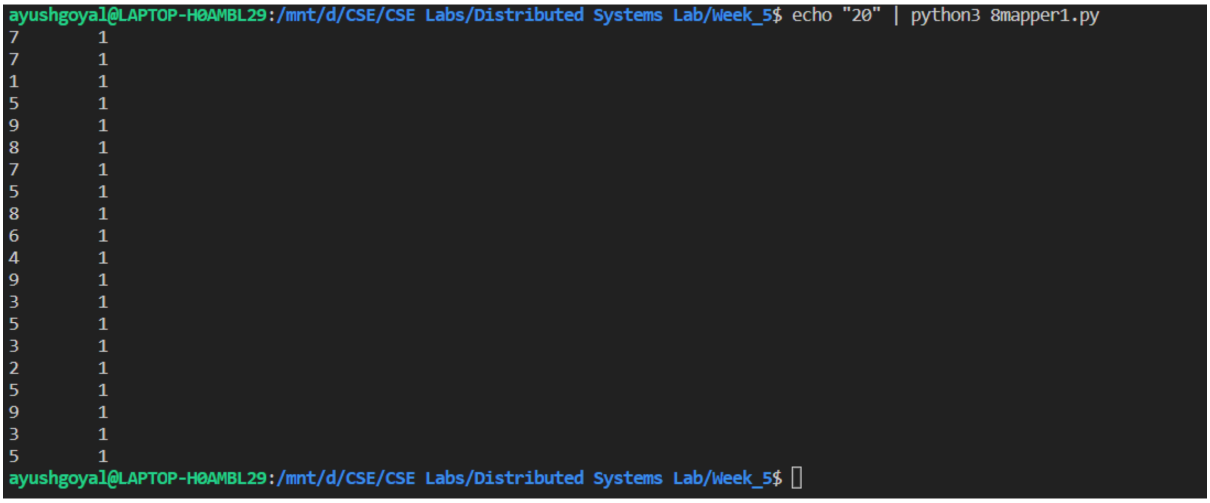
```
import sys
for line in sys.stdin:
        number, count = line.strip().split('\t', 1)
        print('%s%s%s' %(count, '\t', number))
```

8reducer2.py

```
import sys
total = [0] * 2
for line in sys.stdin:
        count, number = line.strip().split('\t', 1)
        total[int(number) % 2] += int(count)
print('Even count:\t', str(total[0]))
print('Odd count:\t', str(total[1]))
```

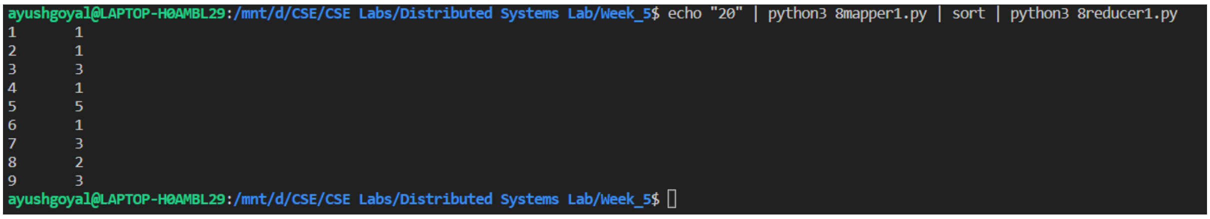
Output:

command: echo "20" | python3 8mapper1.py



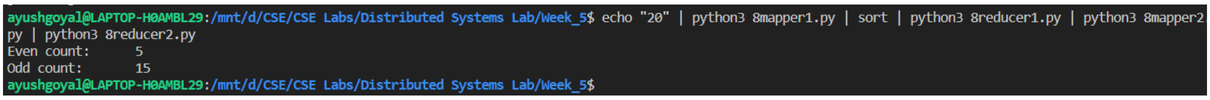
```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ echo "20" | python3 8mapper1.py
7       1
7       1
1       1
5       1
9       1
8       1
7       1
5       1
8       1
6       1
4       1
9       1
3       1
5       1
3       1
2       1
5       1
9       1
3       1
5       1
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ 
```

command: echo "20" | python3 8mapper1.py | sort | python3 8reducer1.py



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ echo "20" | python3 8mapper1.py | sort | python3 8reducer1.py
1       1
2       1
3       3
4       1
5       5
6       1
7       3
8       2
9       3
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ 
```

command: echo "20" | python3 8mapper1.py | sort | python3 8reducer1.py | python3 8mapper2.py | python3 8reducer2.py



```
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$ echo "20" | python3 8mapper1.py | sort | python3 8reducer1.py | python3 8mapper2.
py | python3 8reducer2.py
Even count:     5
Odd count:      15
ayushgoyal@LAPTOP-H0AMBL29:/mnt/d/CSE/CSE Labs/Distributed Systems Lab/Week_5$
```

THE END