SANSKAAR PATNI
180905134 CSE C 23
DS LAB 5
MAP REDUCE

1. Finding word count for column 3 in **heart_disease_data.csv** dataset:-

inpFile.py(Helper file) - takes .csv or .txt file as command line argument and a column index and passes that column index contents to a mapper file

```
import pandas as pd
import sys

file_name = sys.argv[1]
col = int(sys.argv[2]) - 1

# creating a data frame
if file_name.endswith('.csv'):
   df = pd.read_csv(file_name)
   print(df.iloc[:, col].to_string(index=False))
elif file_name.endswith('.txt'):
   with open(file_name, 'r') as f:
       for row in f.readlines():
           count = 0

           for cell in row.split('\t'):
               if count == col:
                   print(cell)
                   break
               count += 1
```

mapper.py

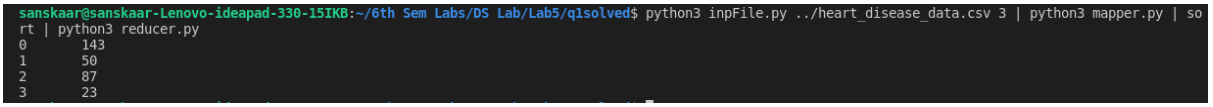
```
import sys
for line in sys.stdin:
   line = line.strip()
   words = line.split()
   for word in words:
       print("%s\t%s" % (word, 1))
```

reducer.py

```
#!/usr/bin/env python
"""reducer.py"""
from operator import itemgetter
import sys
```

```
current_word = None
current_count = 0
word = None
for line in sys.stdin:
    line = line.strip()
    word, count = line.split('\t', 1)
    try:
        count = int(count)
    except ValueError:
        continue
    if current_word == word:
        current_count += count
    else:
        if current_word:
            print('%s\t%s' % (current_word, current_count))
        current_count = count
        current_word = word
if current_word == word:
    print('%s\t%s' % (current_word, current_count))
```

Output Screenshot:-



2. MapReduce program to find maximum frequent words in **German Credit** dataset for DurationOfCreditInMonths column

Converted xlsx to csv

inpFile.py(Helper file) - takes .csv or .txt file as command line argument and a column index and passes that column index contents to a mapper file

```
import pandas as pd
import sys

file_name = sys.argv[1]
col = int(sys.argv[2]) - 1

# creating a data frame
if file_name.endswith('.csv'):
    df = pd.read_csv(file_name)
    print(df.iloc[:, col].to_string(index=False))
elif file_name.endswith('.txt'):
    with open(file_name, 'r') as f:
        for row in f.readlines():
            count = 0
```

```
            for cell in row.split('\t'):
                if count == col:
                    print(cell)
                    break
                count += 1
```

freqmap1.py

```
from __future__ import print_function
import sys

for line in sys.stdin:
   L = [(word.strip().lower(), 1) for word in line.strip().split()]
   for word, n in L:
       print('%s\t%d' % (word, n))
```

freqred1.py

```
#!/usr/bin/env python
# reducer.py
from __future__ import print_function
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))
```

freqmap2.py

```
import sys
for line in sys.stdin:

    word, count = line.strip().split('\t', 1)
    count = int(count)
    print('%d\t%s' % (count, word))
```

freqred2.py

```
from __future__ import print_function
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 after applying freqmap1.py, sort and freqred1.py:

```
sanskaar@sanskaar-Lenovo-ideapad-330-15IKB:~/6th Sem Labs/DS Lab/Lab5/q2$ python3 inpFile.py ../German\ Credit.csv 3 | python3 freqmap1.py | sort | pyt
hon3 freqred1.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
40      1
4       6
42      11
45      5
47      1
48      48
5       1
54      2
60      13
6       75
7       5
72      1
8       7
9       49
```

Output after applying freqmap1.py, sort and freqred1.py followed by freqmap2,py and freqred2.py :

```
sanskaar@sanskaar-Lenovo-ideapad-330-15IKB:~/6th Sem Labs/DS Lab/Lab5/q2$ python3 inpFile.py ../German\ Credit.csv 3 | python3 freqmap1.py | sort | pyt
hon3 freqred1.py| python3 freqmap2.py |python3 freqred2.py
24      184
sanskaar@sanskaar-Lenovo-ideapad-330-15IKB:~/6th Sem Labs/DS Lab/Lab5/q2$
```

3. MapReduce 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 on **example.txt** dataset using location and cost column.
   Finding count and summary (sum=cost column)

   itemmap.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))
```

   itemred.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:

```
sanskaar@sanskaar-Lenovo-ideapad-330-15IKB:~/6th Sem Labs/DS Lab/Lab5/q3$ cat ../example.txt |python3 itemmap.py |sort| python3 itemred.py
50      12268.159999999996
sanskaar@sanskaar-Lenovo-ideapad-330-15IKB:~/6th Sem Labs/DS Lab/Lab5/q3$
```

4. Write a mapper and reducer program for word count by defining a separator (>)instead of using “\t”.
   Using **example.txt** dataset and **covid_19_data.csv**

   inpFile.py(Helper file) - takes .csv or .txt file as command line argument and a column index and passes that column index contents to a mapper file

```
import pandas as pd
import sys
```

```
file_name = sys.argv[1]
col = int(sys.argv[2]) - 1

# creating a data frame
if file_name.endswith('.csv'):
   df = pd.read_csv(file_name)
   print(df.iloc[:, col].to_string(index=False))
elif file_name.endswith('.txt'):
   with open(file_name, 'r') as f:
       for row in f.readlines():
           count = 0

           for cell in row.split('\t'):
               if count == col:
                   print(cell)
                   break
               count += 1
```

sepmap.py

```
import sys


def read_input(file):
   for line in file:
       yield line.split()


def main(separator='\t'):
   data = read_input(sys.stdin)
   separator = sys.argv[1]
   for words in data:
       for word in words:
           print('%s%s%d' % (word, separator, 1))


if __name__ == "__main__":
   main()
```

sepred.py

```
from itertools import groupby
from operator import itemgetter
from os import sep
```

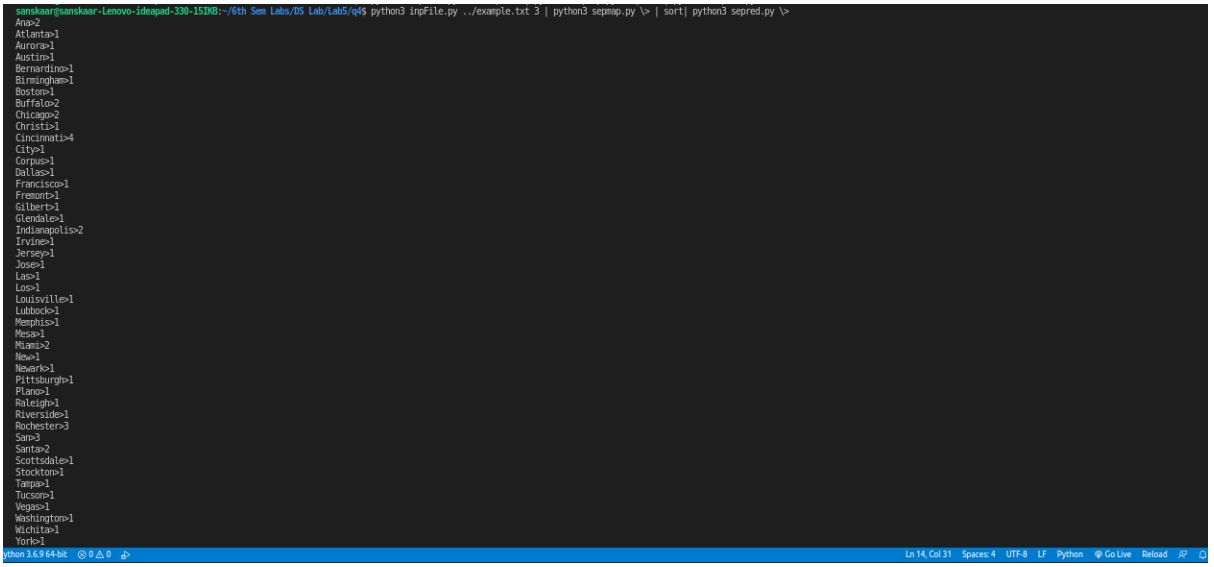
```python
import sys


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


def main(separator='\t'):
    if(sys.argv[1] != ""):
        separator = sys.argv[1]
        data = read_mapper_output(sys.stdin, separator=separator)
    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, separator,
total_count))
        except ValueError:
            pass


if __name__ == "__main__":
    main()
```

Using separator >
Output after sepmap.py and sorting for example.txt dataset:

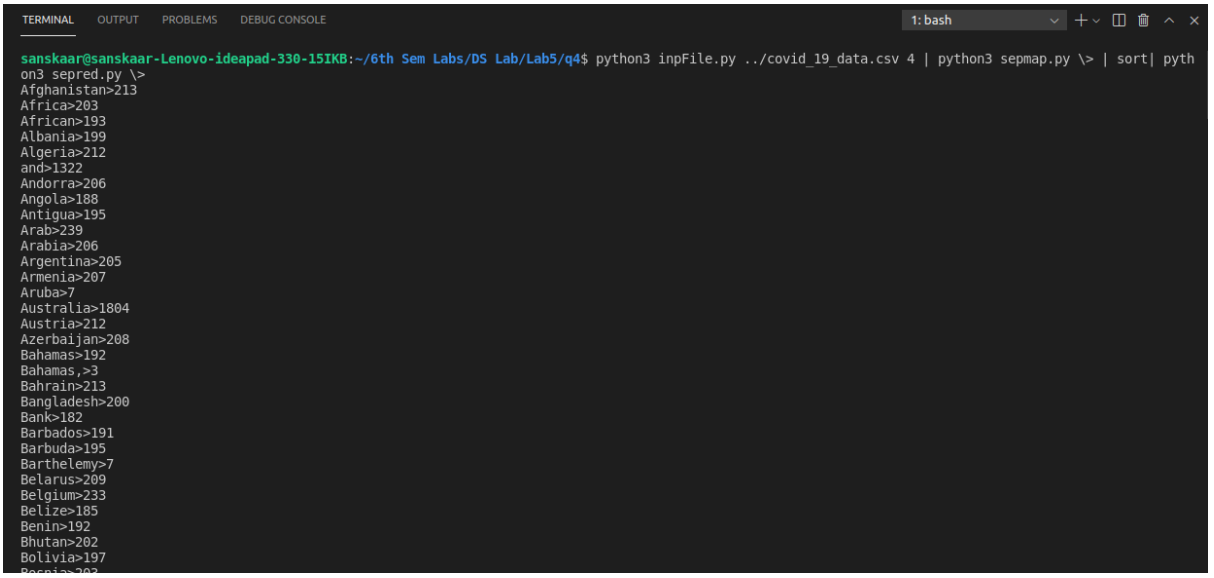
```
sanskaar@sanskaar-Lenovo-ideapad-330-15IKB:~/6th Sem Labs/DS Lab/Lab5/q4$ python3 inpFile.py ../example.txt 3 | python3 sepmap.py \> | sort
Ana>1
Ana>1
Atlanta>1
Aurora>1
Austin>1
Bernardino>1
Birmingham>1
Boston>1
Buffalo>1
Buffalo>1
Chicago>1
Chicago>1
Christi>1
Cincinnati>1
Cincinnati>1
Cincinnati>1
Cincinnati>1
City>1
Corpus>1
Dallas>1
Francisco>1
Fremont>1
Gilbert>1
Glendale>1
Indianapolis>1
Indianapolis>1
Irvine>1
Jersey>1
Jose>1
Las>1
Los>1
Louisville>1
Lubbock>1
Memphis>1
Mesa>1
Miami>1
Miami>1
```

```
Miami>1
Miami>1
New>1
Newark>1
Pittsburgh>1
Plano>1
Raleigh>1
Riverside>1
Rochester>1
Rochester>1
Rochester>1
San>1
San>1
San>1
Santa>1
Santa>1
Scottsdale>1
Stockton>1
Tampa>1
Tucson>1
Vegas>1
Washington>1
Wichita>1
York>1
sanskaar@sanskaar-Lenovo-ideapad-330-15IKB:~/6th Sem Labs/DS Lab/Lab5/q4$
```

Output after sepmad,sort,sepred.py example.txt dataset:



Output for covid_19_data.csv:(not complete output since output is too big)



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**

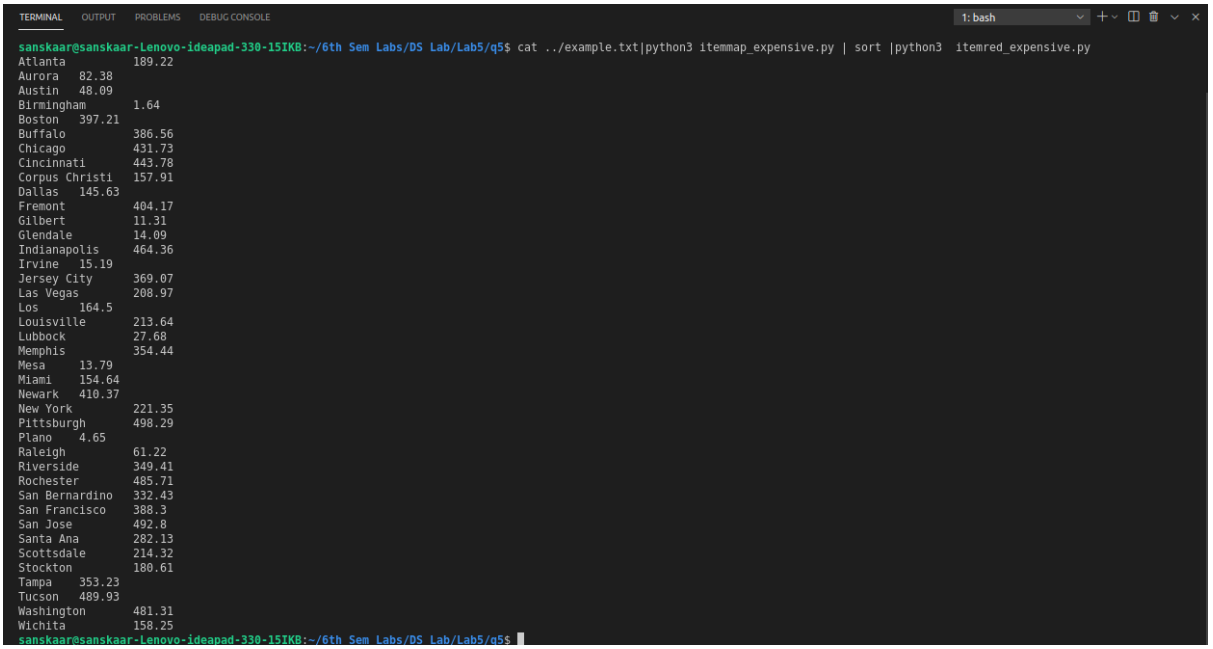
itemmap_expensive.py

```
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))
```

itemred_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

    old_key = current_key
    if float(current_value) > float(max_value):
        max_value = float(current_value)

if old_key != None:
    print(old_key, "\t", max_value)
```

Output:


TERMINAL
OUTPUT
PROBLEMS
DEBUG CONSOLE
1: bash
sanskaar@sanskaar-Lenovo-ideapad-330-15IKB:~/6th Sem Labs/DS Lab/Lab5/q5$ cat ../example.txt|python3 itemmap_expensive.py | sort |python3 itemred_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
Newark 410.37
New York 221.35
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
sanskaar@sanskaar-Lenovo-ideapad-330-15IKB:~/6th Sem Labs/DS Lab/Lab5/q5$