Nalongsone Danddank Student ID : 14958950 StarID: jf3893pd
Email: nalongsone.danddank@my.metrostate.edu\

ICS 432 - 01 — Distributed and Cloud Computing Fall 2021

Assignment #3 : MapReduce

Exercise 1: Analyzing Stock Data.

Task 1.1:

1- Print out some line for looking for some keys, stock_symbol in the csv file and find out some duplicate of the key that should be the value. Then, implement python code mapper to print out all of key, stock_symbol and follow by 1. finally, implement python code for reduce that combine all value that are same key, and print out the key with total counting number.

For example: output key and value of reduce, GA 12, GAB 75, GAH 18, GAI 34, GAJ 27...

2-

mapper1.py > ...

```python
#!/usr/bin/env python

import sys

for line in sys.stdin:
    line = line.strip()
    words = line.split(',')
    print("%s\t%s" % (words[1], 1))

```

3-

reducer1.py > ...

```python
#!/usr/bin/env python

import sys

current_word = None
current_count = 0
word = None

for line in sys.stdin:
    line = line.strip()
    (word, count) = line.split('\t')
    count = int(count)
    if current_word == word:
        current_count += 1
    else:
        if current_word:
            print(current_word, current_count)
        current_count = count
        current_word = word

if current_word == word:
    print(current_word, current_count)

```

4-

```
GYC     1
GYC     1
GYC     1
GYC     1
GYC     1
GYC     1
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper1.py | sort
 | python3 reducer1.py
GA 12
GAB 75
GAH 18
GAI 34
GAJ 27
GAM 88
GAP 76
GAR 1
GAS 70
GAT 6
GB 28
GBB 9
GBE 20
GBF 10
GBL 41
GBX 55
GCA 12
GCF 8
GCH 54
GCI 54
GCO 88
GCS 14
GCV 43
GD 100
GDF 48
GDI 55
GDL 8
GDO 1
GDP 71
GDV 15
GE 155
GEA 5
GEC 22
GED 22
GEF 46
GEG 12
GEJ 5
GEO 55
GEP 14
GER 11
GES 45
GET 61
GEX 7
GF 64
GFA 8
GFF 54
GFI 52
GFW 25
GFY 16
GFZ 25
```

Task 1.2:

1- Print out some line for looking for some keys, stock_symbol in the csv file and find out some the same key that should be the value. Then, implement python code mapper to print out all of key, stock_symbol and follow by its value, stock_price_high. finally, implement python code for reduce that compare all value that are same key to get the highest, and print out the key and the highest stock price.

For example: output key and value of reduce, GA 13.35, GAB 11.99, GAH 25.3, GAI 14.62, GAJ 25.79...

2-

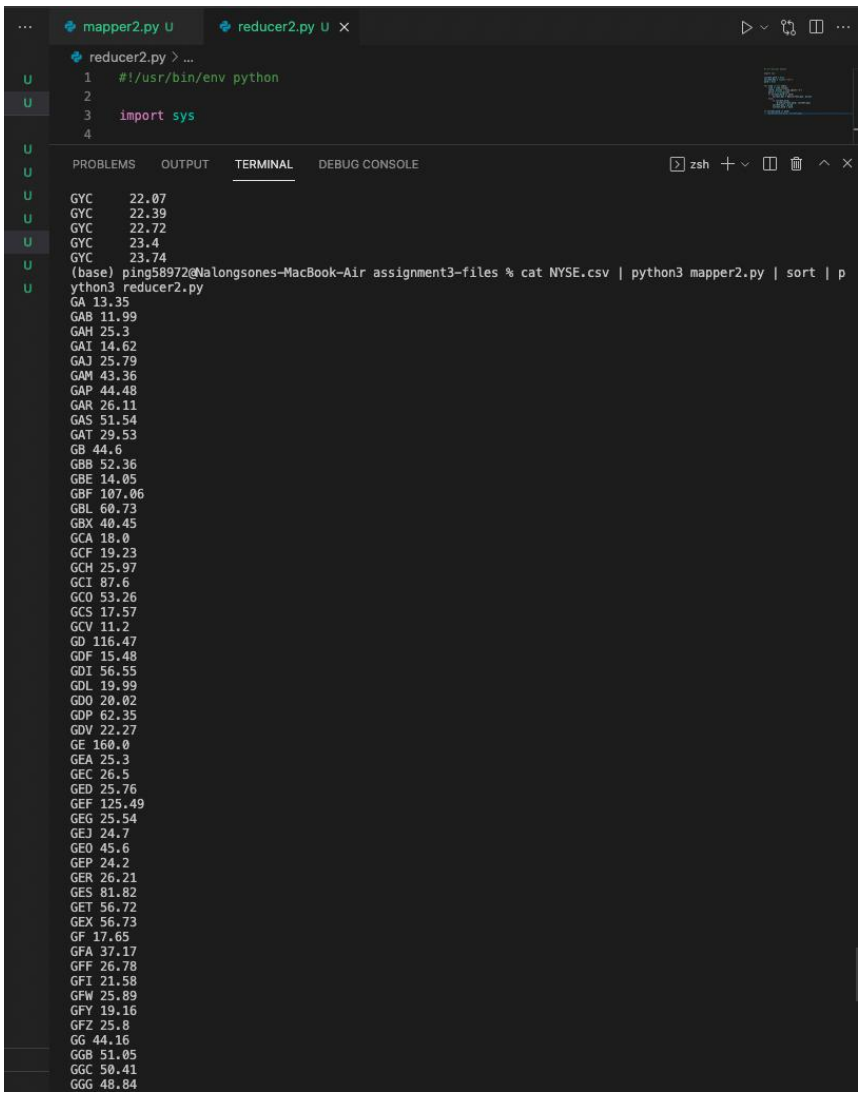
3-

mapper2.py U reducer2.py U

reducer2.py > ...

```python
#!/usr/bin/env python

import sys

current_word = None
current_max = float('-inf')
word = None

for line in sys.stdin:
    line = line.strip()
    (word, price) = line.split('\t')
    price = float(price)
    if current_word == word:
        current_max = max(current_max, price)
    else:
        if current_word:
            print(current_word, current_max)
        current_max = price
        current_word = word

if current_word == word:
    print(current_word, current_max)
```

4-



Task 1.3:

1- Print out some line for looking for some keys, stock_symbol in the csv file and find out some the same key and its price and its volume . Then, implement python code mapper to print out all of key, stock_symbol and follow by its values, stock_price_high and its volume greater than 250,000. finally, implement python code for reduce that compare all value that are same key to get the highest, and print out the key and the highest stock price and volume>250000.

For example: output key and value of reduce, GA 13.35 299100, GAB 10.6 284400, GAI 4.75 256400, GAP 32.9 371200, GAS 51.54 421800...

2-

mapper3.py U | reducer3.py U | NYSE.csv U

mapper3.py > ...

```python
#!/usr/bin/env python

import sys

for line in sys.stdin:
    line = line.strip()
    words = line.split(',')
    if int(words[7]) > 250000:
        print("%s\t%s\t%s" % (words[1], words[4], words[7]))

```

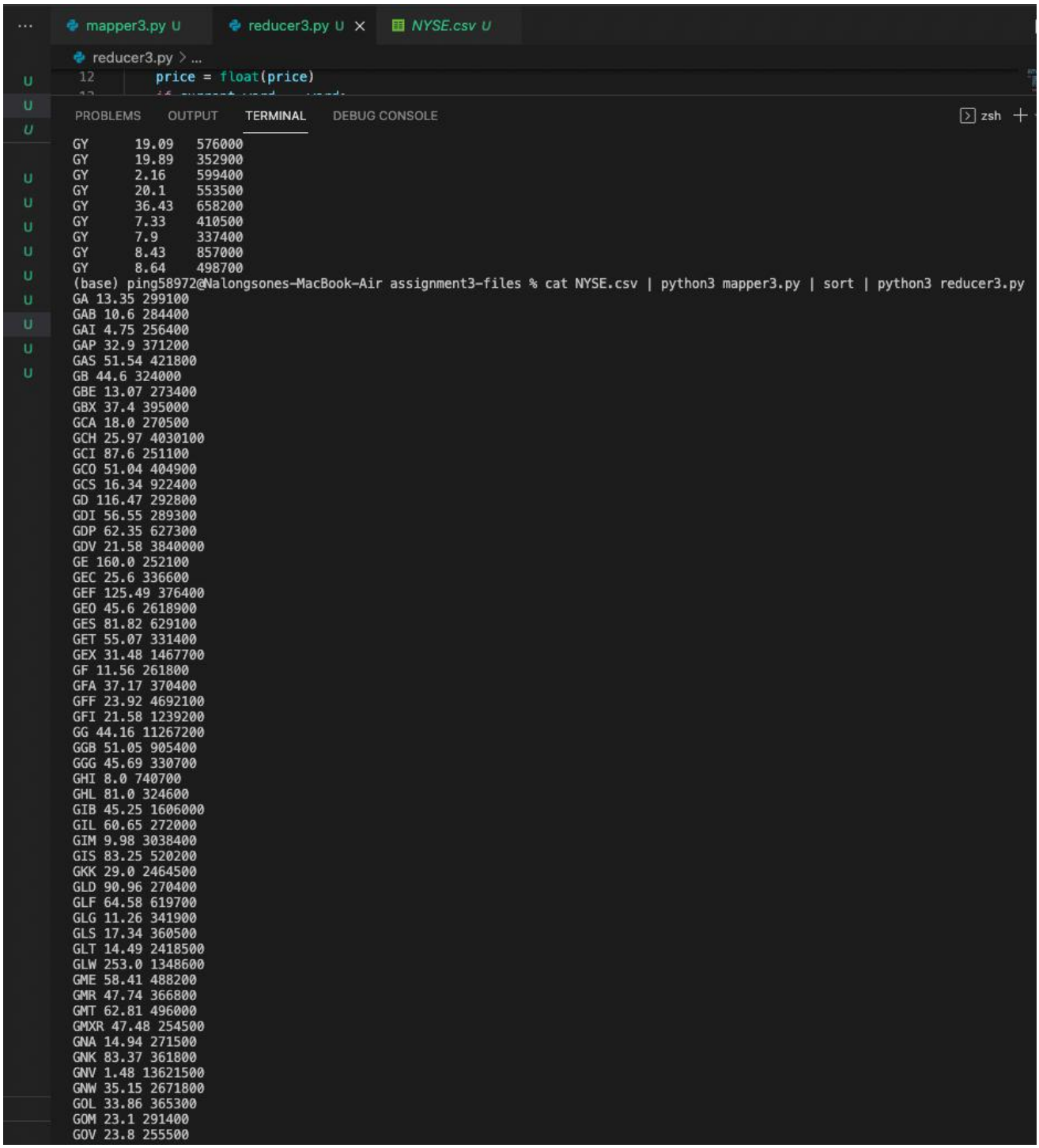
3-

mapper3.py U | reducer3.py U | NYSE.csv U

reducer3.py > ...

```python
#!/usr/bin/env python

import sys

current_word = None
current_max = float('-inf')
word = None

for line in sys.stdin:
    line = line.strip()
    (word, price, volume) = line.split('\t')
    price = float(price)
    if current_word == word:
        current_max = max(current_max, price)
    else:
        if current_word:
            print(current_word, current_max, volume)
        current_max = price
        current_word = word

if current_word == word:
    print(current_word, current_max, volume)

```

4-



Task 1.4:

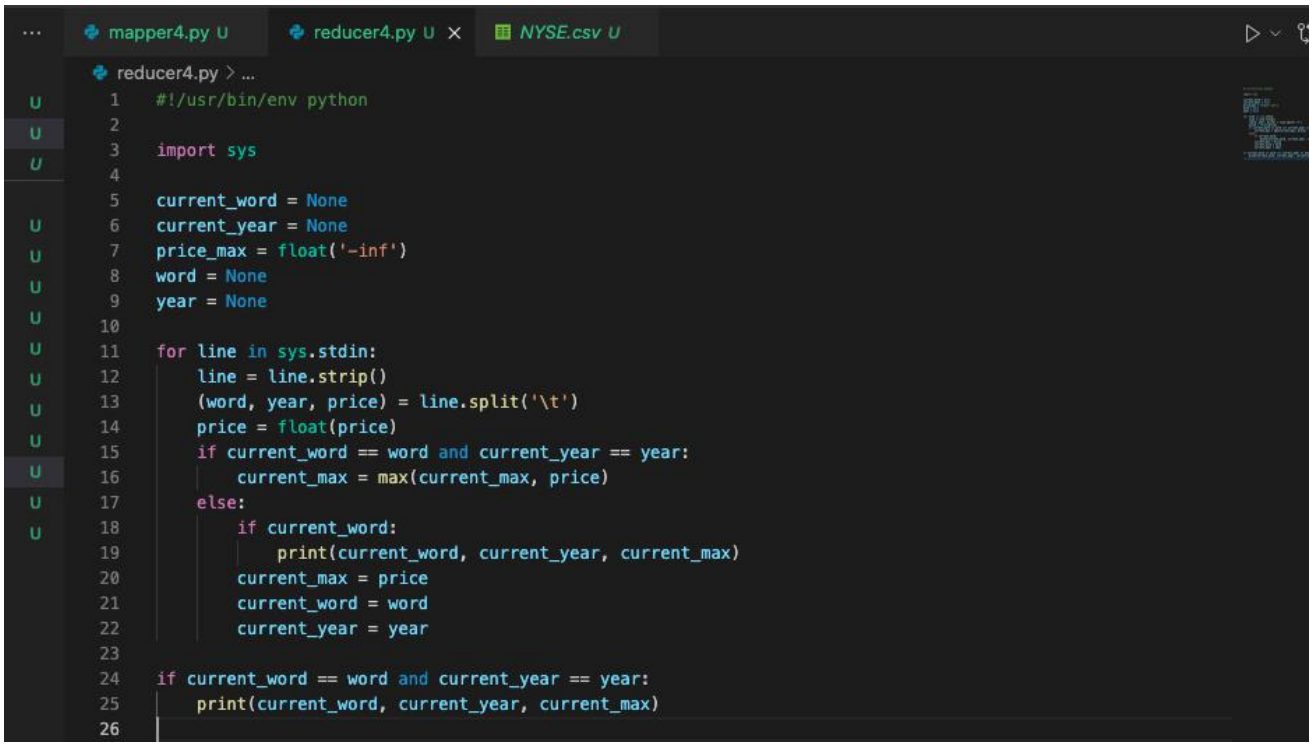
1- Print out some line for looking for some keys, stock_symbol in the csv file and find out some the same keys, stock and date, and its price. Then, implement python code mapper to print out all of keys, stock_symbol with year which extract by using datetime, and follow by its values, stock_price_high. finally, implement python code for reduce that compare all price that are same keys, stock and year to get the highest price, and print out each stock and each year with the highest stock price.

For example: output key and value of reduce, GYC 2019 18.15, GYC 2018 20.6, GYB 2020 21.41, GY 2019 8.43, GY 2018 8.64...

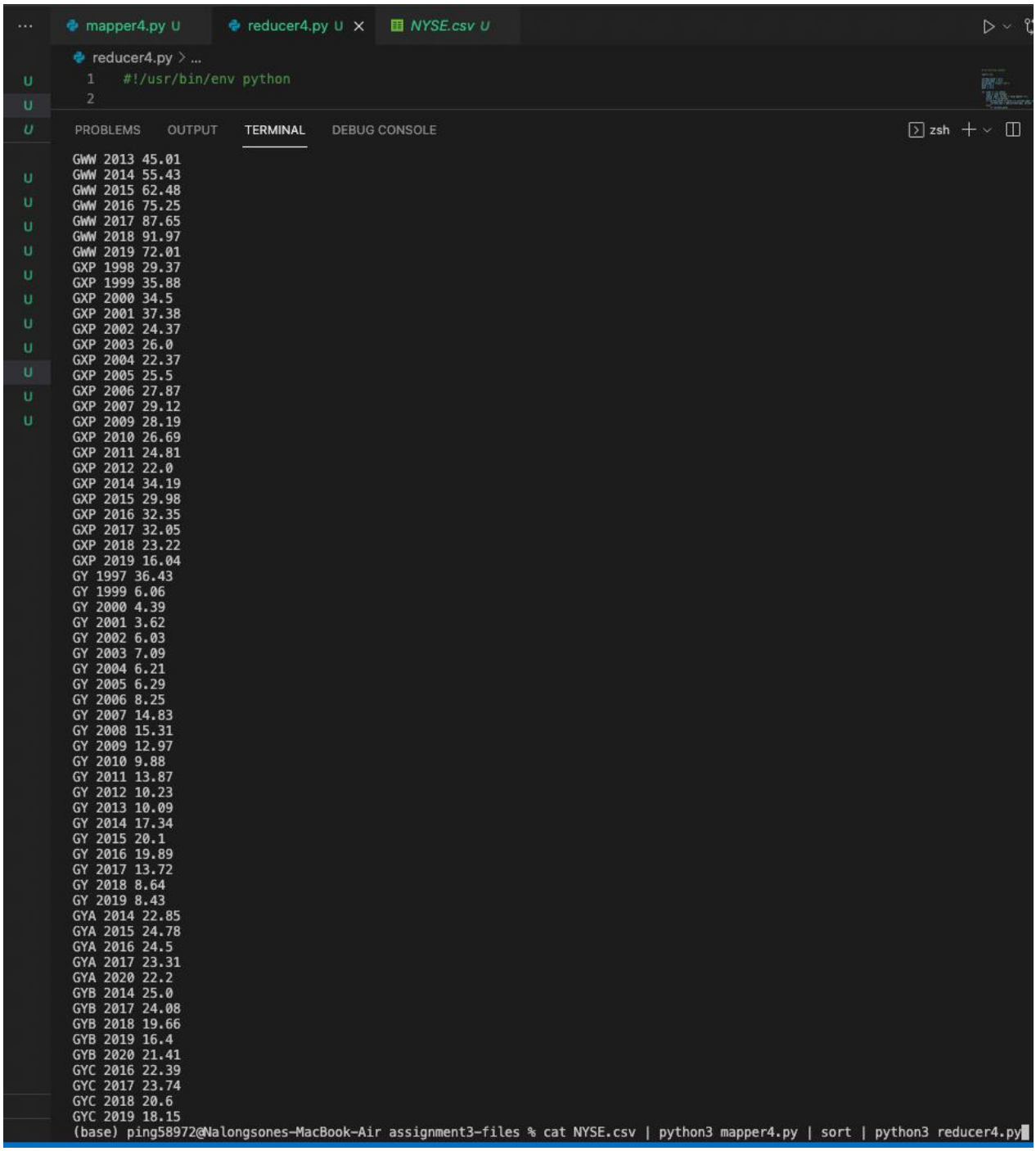
2-


mapper4.py U
reducer4.py U
NYSE.csv U
mapper4.py > ...
#!/usr/bin/env python
import sys
from datetime import datetime
for line in sys.stdin:
line = line.strip()
words = line.split(',')
date = datetime.strptime(words[2], "%m/%d/%Y")
year = date.strftime('%Y')
print("%s\t%s\t%s" % (words[1], year, words[4]))

3-


mapper4.py U
reducer4.py U
NYSE.csv U
reducer4.py > ...
#!/usr/bin/env python
import sys
current_word = None
current_year = None
price_max = float('-inf')
word = None
year = None
for line in sys.stdin:
line = line.strip()
(word, year, price) = line.split('\t')
price = float(price)
if current_word == word and current_year == year:
current_max = max(current_max, price)
else:
if current_word:
print(current_word, current_year, current_max)
current_max = price
current_word = word
current_year = year
if current_word == word and current_year == year:
print(current_word, current_year, current_max)

4-



Task 1.5:

1- First do the Job 1 by run the CSV file with mapper1.py and reducer1.py, then go to do Job 2 by taking the output to run the mapper5.py and reducer5.py which implement to count all number that output from Job 1 to find total number distinct stock symbols.

Output: 1 154

2-

mapper1.py

mapper5.py

```python
#!/usr/bin/env python

import sys

for line in sys.stdin:
    print("%s\t%s" % (1, 1))
```

3-

reducer1.py

```python
#!/usr/bin/env python

import sys

current_word = None
current_count = 0
word = None

for line in sys.stdin:
    line = line.strip()
    (word, count) = line.split('\t')
    count = int(count)
    if current_word == word:
        current_count += 1
    else:
        if current_word:
            print(current_word, current_count)
        current_count = count
        current_word = word

if current_word == word:
    print(current_word, current_count)
```

reducer5.py

```python
#!/usr/bin/env python

import sys

key_ = "1"
count_ = 0

for line in sys.stdin:
    line = line.strip()
    (key, value) = line.split('\t')
    count = int(value)
    count_ += count

print(key_, count_)
```

4-

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE                                zsh

1       1
1       1
1       1
1       1
1       1
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper1.py | sort | python3 reducer1.py | python
3 mapper5.py | python3 reducer5.py
1 154
(base) ping58972@Nalongsones-MacBook-Air assignment3-files %
```

Exercise 2: Implementing Relational Union and Intersection using mapreduce.

Task 2.1:

1- Print out some line for looking for some product ID in both the txt file and find out some duplicate of the ID. Then, implement python code mapper to print out and sort all of both file each line. finally, implement python code for reduce ignore the same product ID, and print out the unique product ID line with their describe and price.

For example output:

1001 Zip Bag 100

1002 Harness 150

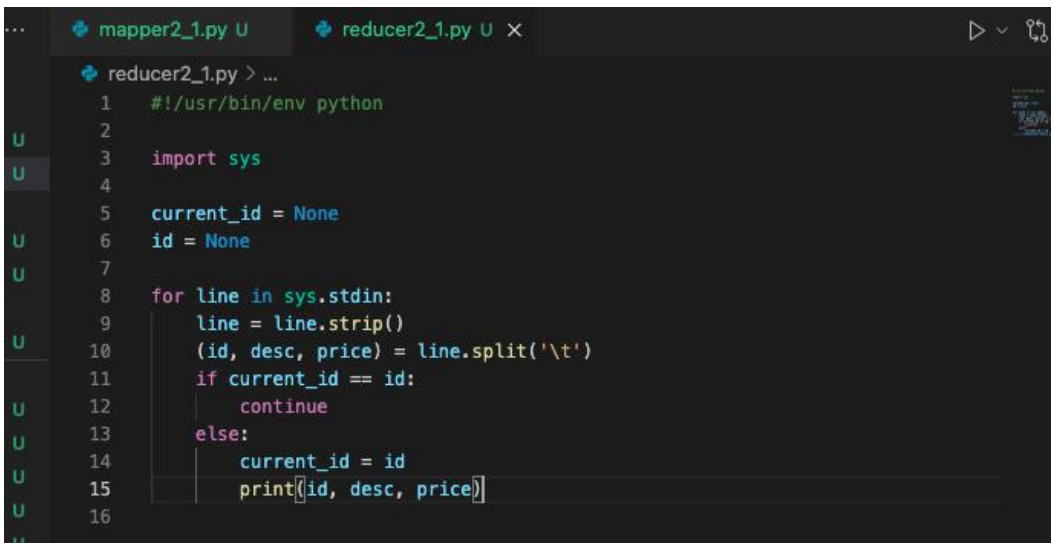
1003 Full Charger 125

1004 Big Helmet 40

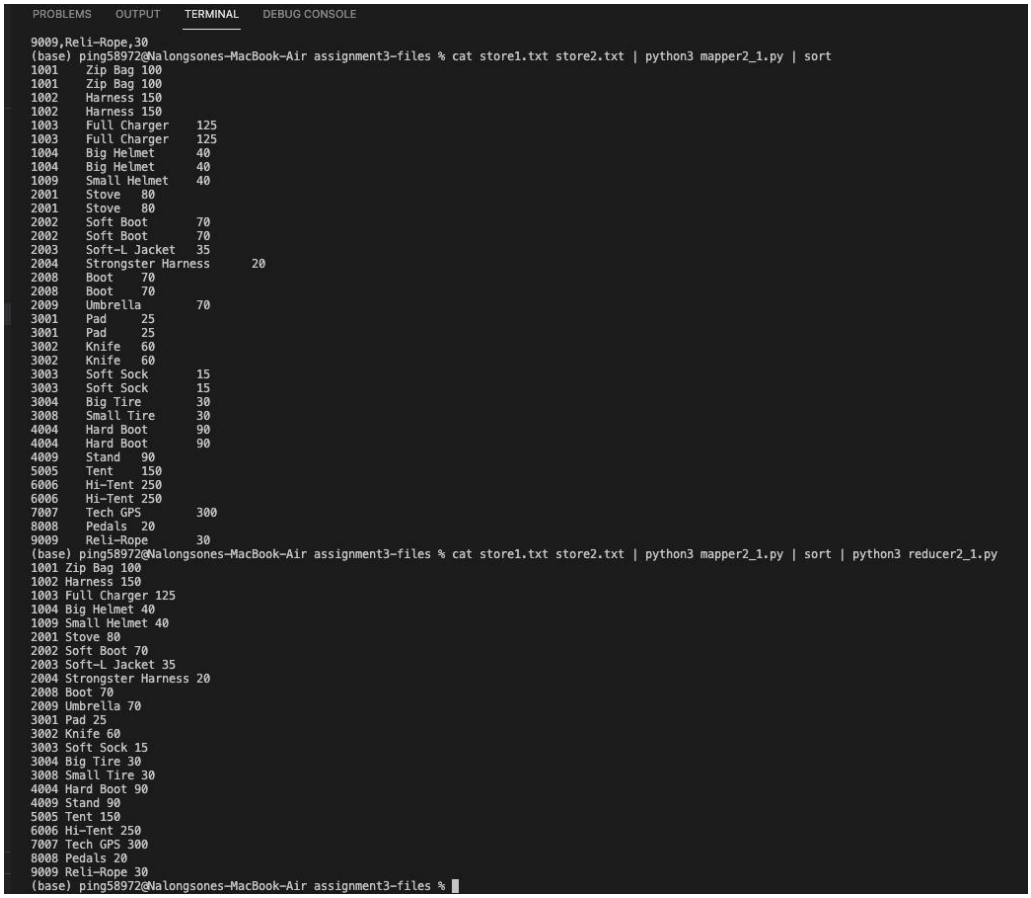
1009 Small Helmet 40

2001 Stove 80
2002 Soft Boot 70
2003 Soft-L Jacket 35
2004 Strongster Harness 20
2008 Boot 70
2009 Umbrella 70
3001 Pad 25
3002 Knife 60
3003 Soft Sock 15
3004 Big Tire 30
3008 Small Tire 30
4004 Hard Boot 90
4009 Stand 90
5005 Tent 150
6006 Hi-Tent 250
7007 Tech GPS 300
8008 Pedals 20
9009 Reli-Rope 302-

3-



4-



Task 2.2:

1- Print out some line for looking for some product ID in both the txt file and find out some duplicate of the ID. Then, implement python code mapper to print out and sort all of both file each line. finally, implement python code for reduce ignore the product ID which only belong in just one file, and print out the unique product ID which belong to both files with their describe and price.

For example output:

1001 Zip Bag 100
1002 Harness 150
1003 Full Charger 125
1004 Big Helmet 40
2001 Stove 80
2002 Soft Boot 70
2008 Boot 70
3001 Pad 25
3002 Knife 60
3003 Soft Sock 15
4004 Hard Boot 90
6006 Hi-Tent 250

2-

mapper2_2.py U × reducer2_2.py U

mapper2_2.py > ...

```python
#!/usr/bin/env python

import sys

for line in sys.stdin:
    line = line.strip()
    words = line.split(',')
    print("%s\t%s\t%s" % (words[0], words[1], words[2]))

```

3-

mapper2_2.py U reducer2_2.py U ×

reducer2_2.py > ...

```python
#!/usr/bin/env python

import sys

current_id = None
id = None

for line in sys.stdin:
    line = line.strip()
    (id, desc, price) = line.split('\t')
    if current_id == id:
        print(id, desc, price)
    else:
        current_id = id

```

4-

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
3008    Small Tire      30
4004    Hard Boot       90
4004    Hard Boot       90
4009    Stand   90
5005    Tent    150
6006    Hi-Tent 250
6006    Hi-Tent 250
7007    Tech GPS        300
8008    Pedals  20
9009    Reli-Rope       30
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat store1.txt store2.txt | python3 mapper2_2.py | sort | python3 reducer2_2.py
1001 Zip Bag 100
1002 Harness 150
1003 Full Charger 125
1004 Big Helmet 40
2001 Stove 80
2002 Soft Boot 70
2008 Boot 70
3001 Pad 25
3002 Knife 60
3003 Soft Sock 15
4004 Hard Boot 90
6006 Hi-Tent 250
(base) ping58972@Nalongsones-MacBook-Air assignment3-files %
```