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

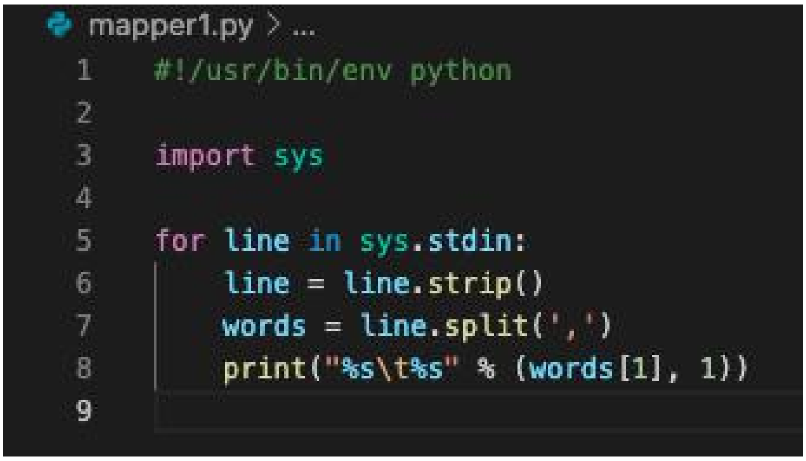
Exercise 1: Analyzing Stock Data.

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...

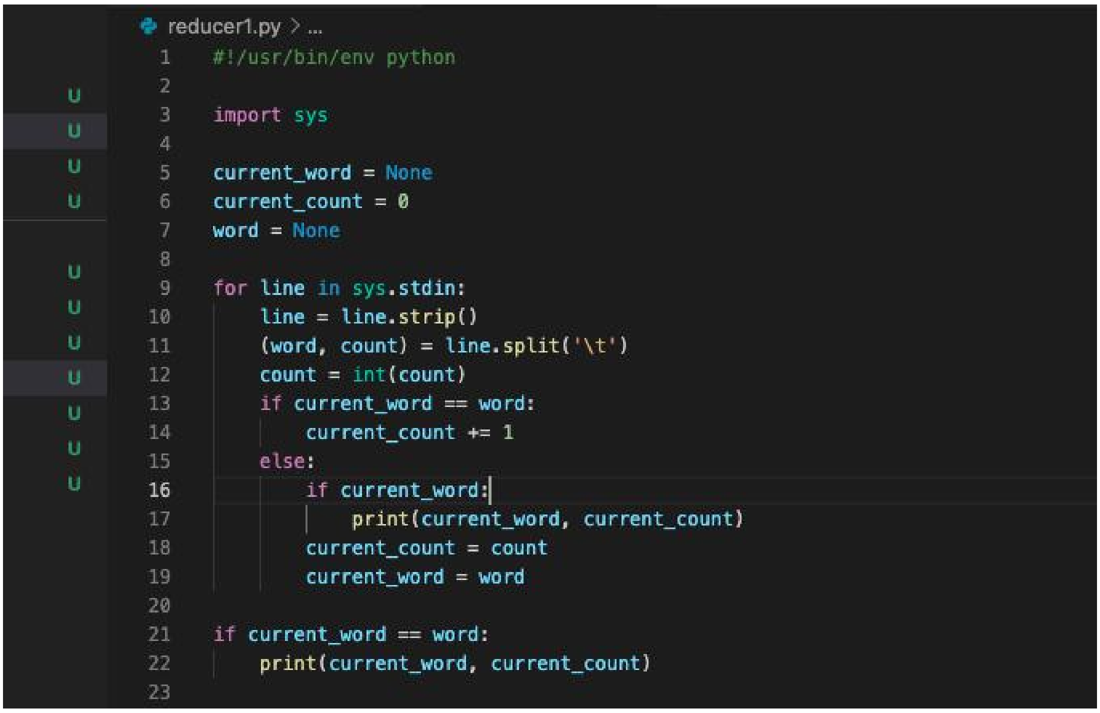
2-



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))

```

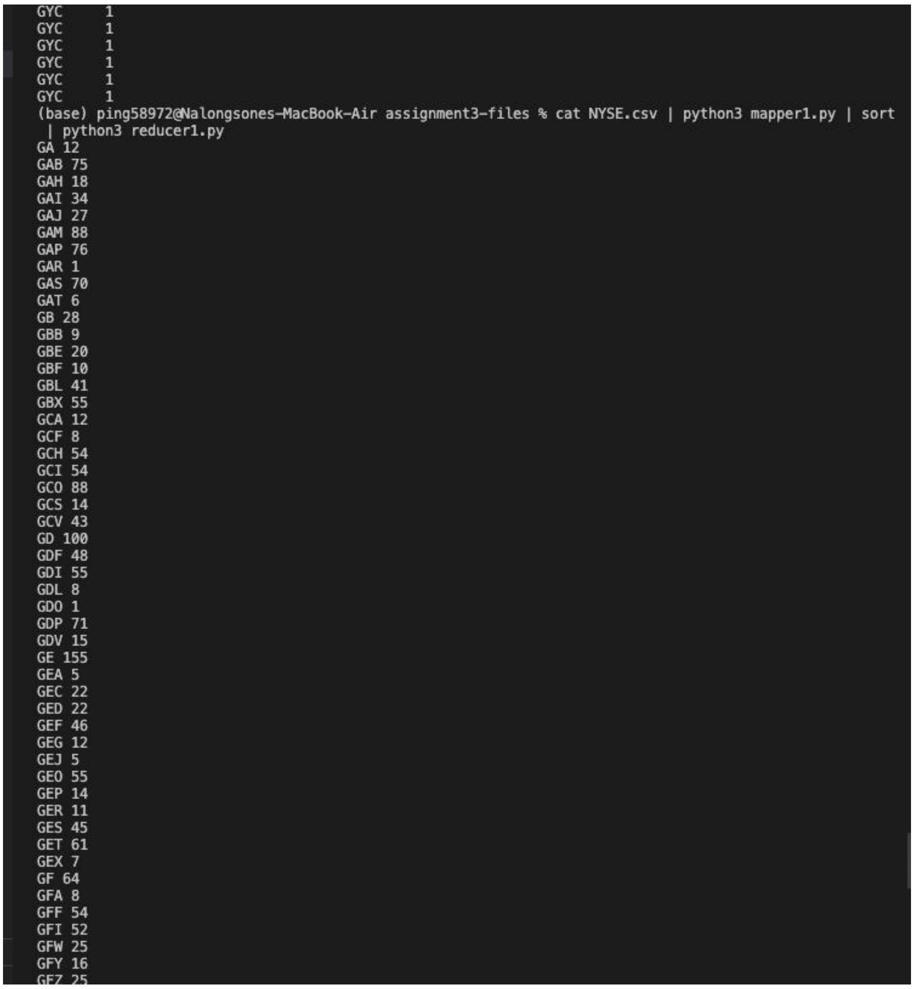
3-



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-



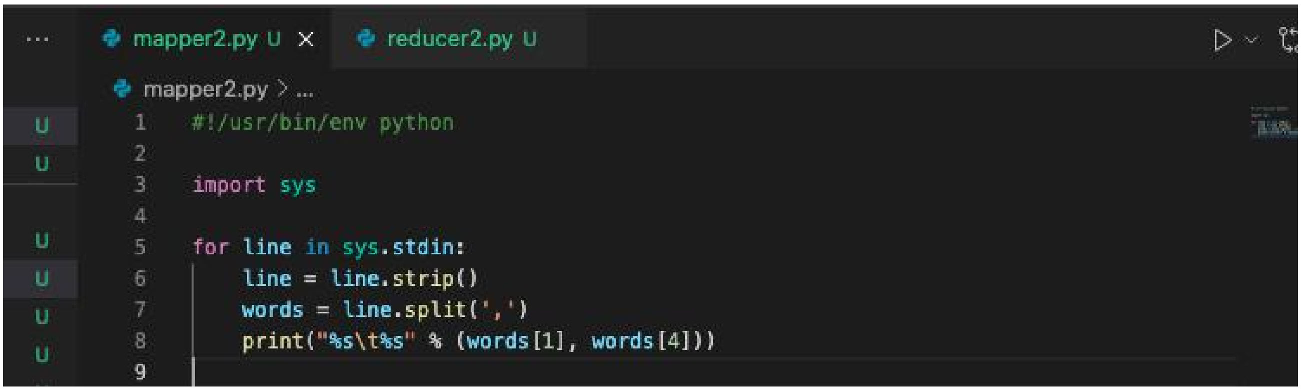
```
GYC     1
GYC     1
GYC     1
GYC     1
GYC     1
GYC     1
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper1.py | sort
 | python3 reducer1.py
GA 12
GAB 75
GAH 18
GAI 34
GAJ 27
GAM 88
GAP 76
GAR 1
GAS 70
GAT 6
GB 28
GBB 9
GBE 20
GBF 10
GBL 41
GBX 55
GCA 12
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.

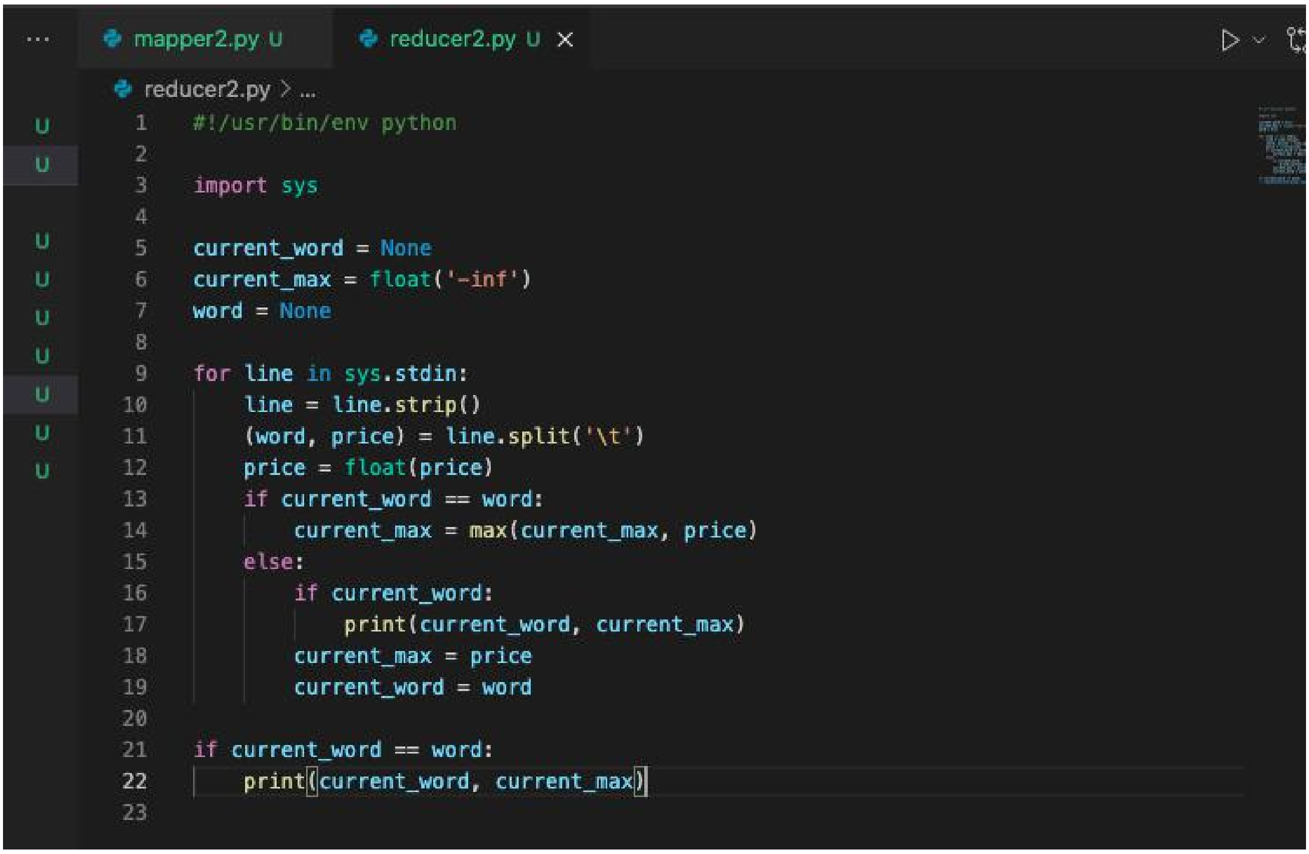
For example: output key and value of reduce, GA 13.35, GAB 11.99, GAH 25.3, GAI 14.62, GAJ 25.79...

2-



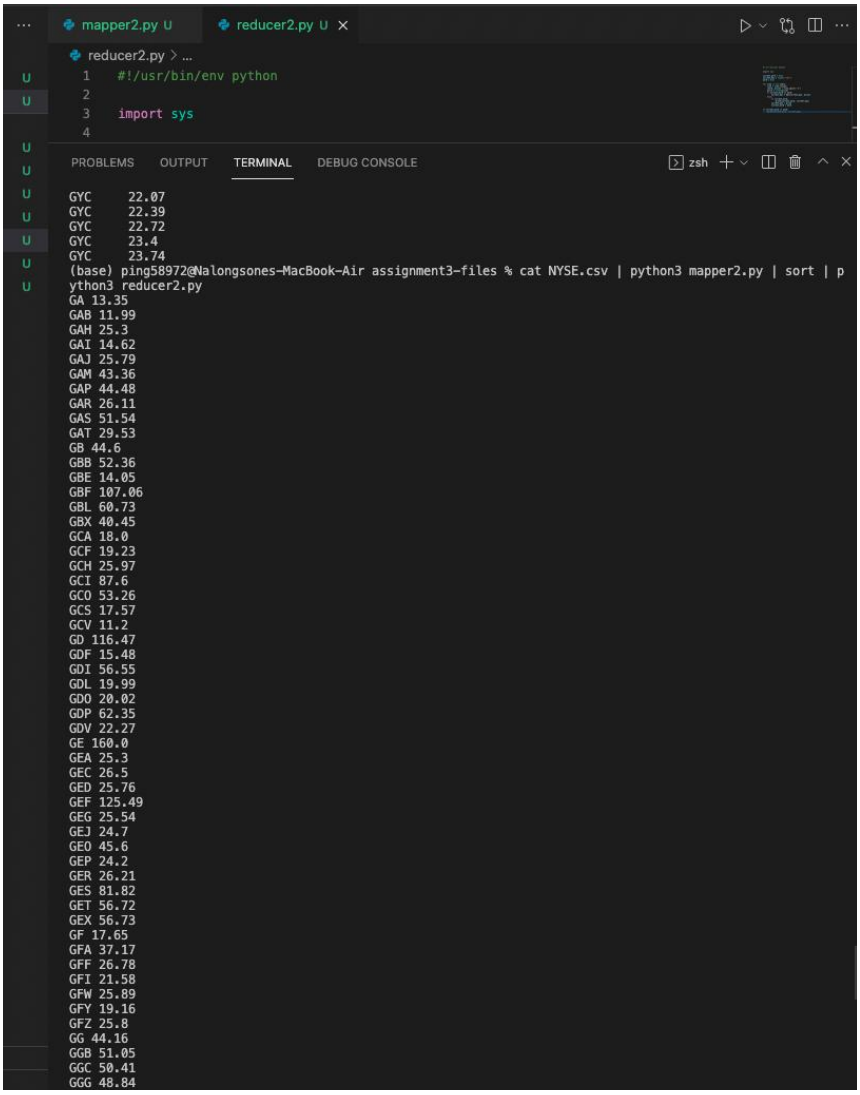
```
#!/usr/bin/env python

import sys

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

```

3-



```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-



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

import sys

```

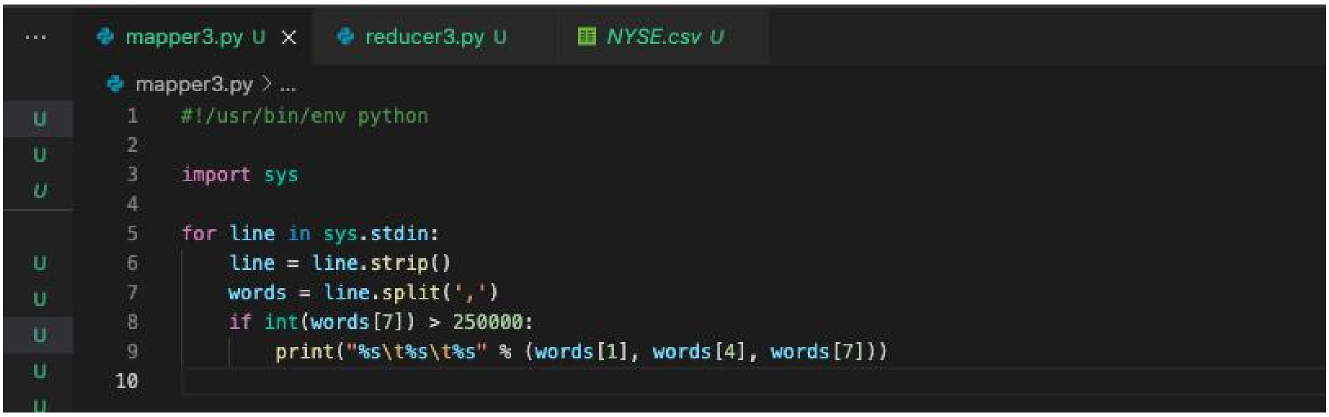
```
GYC     22.07
GYC     22.39
GYC     22.72
GYC     23.4
GYC     23.74
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper2.py | sort | p
ython3 reducer2.py
GA 13.35
GAB 11.99
GAH 25.3
GAI 14.62
GAJ 25.79
GAM 43.36
GAP 44.48
GAR 26.11
GAS 51.54
GAT 29.53
GB 44.6
GBB 52.36
GBE 14.05
GBF 107.06
GBL 60.73
GBX 40.45
GCA 18.0
GCF 19.23
GCH 25.97
GCI 87.6
GCO 53.26
GCS 17.57
GCV 11.2
GD 116.47
GDF 15.48
GDI 56.55
GDL 19.99
GDO 20.02
GDP 62.35
GDV 22.27
GE 160.0
GEA 25.3
GEC 26.5
GED 25.76
GEF 125.49
GEG 25.54
GEJ 24.7
GEO 45.6
GEP 24.2
GER 26.21
GES 81.82
GET 56.72
GEX 56.73
GF 17.65
GFA 37.17
GFF 26.78
GFI 21.58
GFW 25.89
GFY 19.16
GFZ 25.8
GG 44.16
GGB 51.05
GGC 50.41
GGG 48.84
```

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.

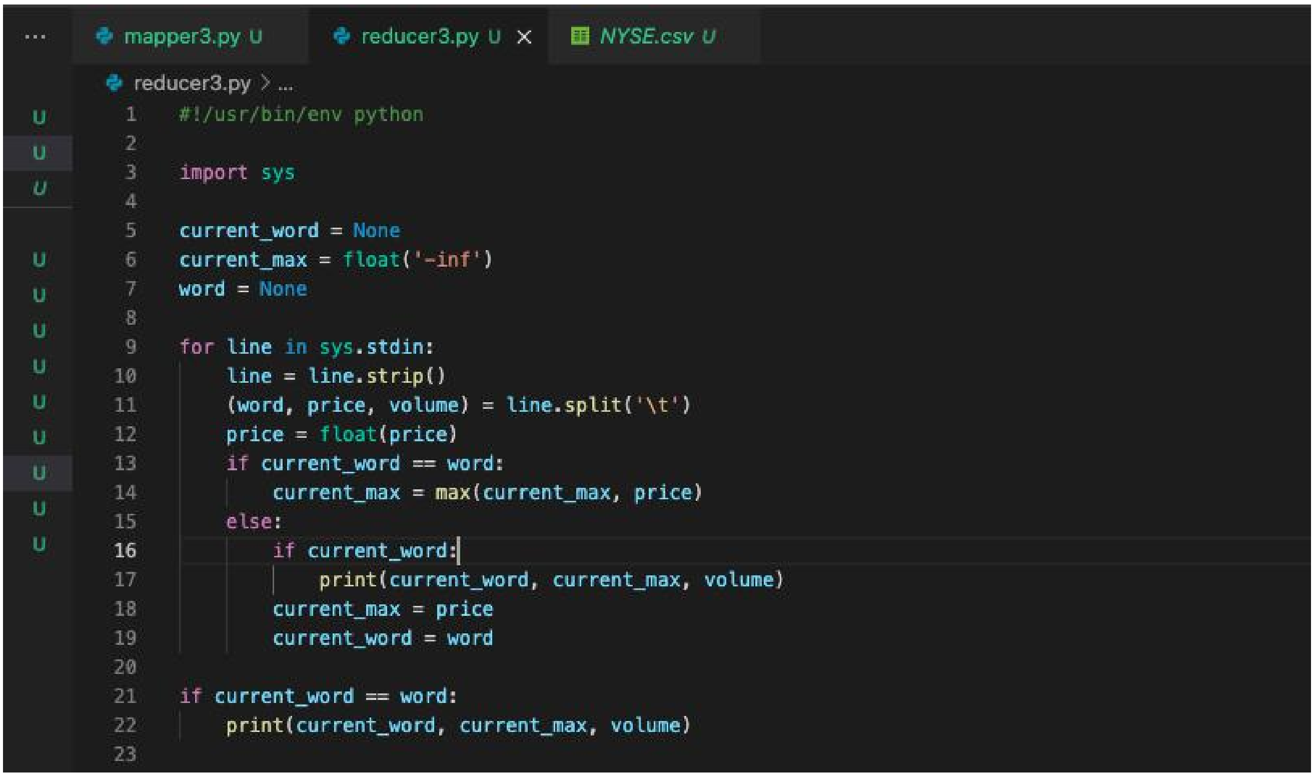
For example: output key and value of reduce, GA 13.35 299100, GAB 10.6 284400, GAI 4.75 256400, GAP 32.9 371200, GAS 51.54 421800...

2-



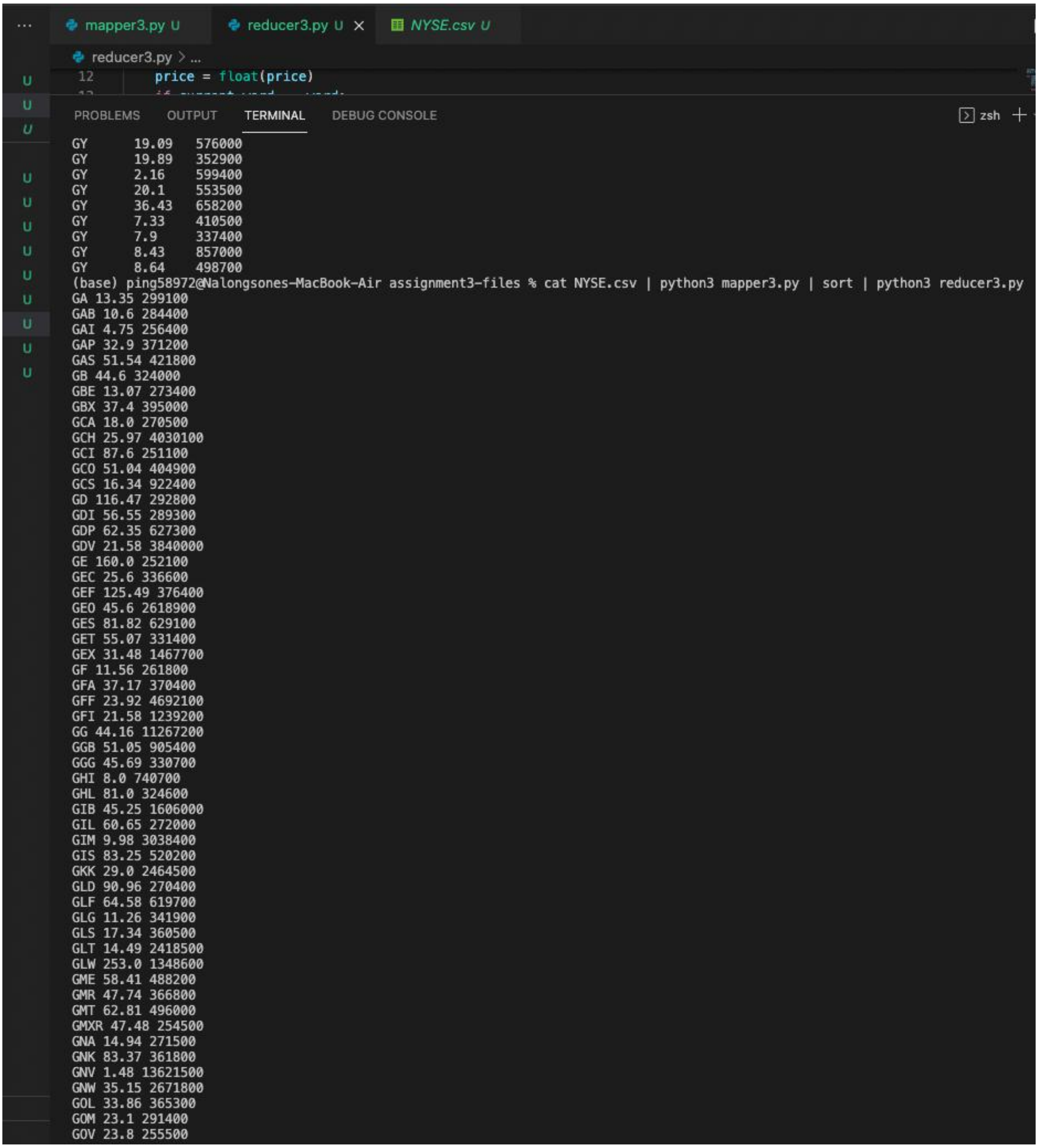
```python
#!/usr/bin/env python

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-



```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-



```
        price = float(price)
```

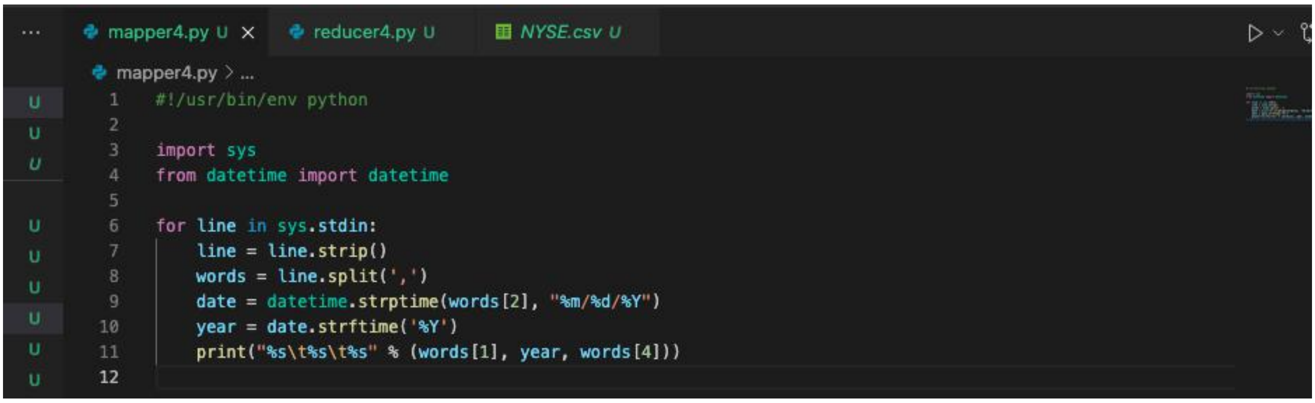
```
GY      19.09   576000
GY      19.89   352900
GY      2.16    599400
GY      20.1    553500
GY      36.43   658200
GY      7.33    410500
GY      7.9     337400
GY      8.43    857000
GY      8.64    498700
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper3.py | sort | python3 reducer3.py
GA 13.35 299100
GAB 10.6 284400
GAI 4.75 256400
GAP 32.9 371200
GAS 51.54 421800
GB 44.6 324000
GBE 13.07 273400
GBX 37.4 395000
GCA 18.0 270500
GCH 25.97 4030100
GCI 87.6 251100
GCO 51.04 404900
GCS 16.34 922400
GD 116.47 292800
GDI 56.55 289300
GDP 62.35 627300
GDV 21.58 3840000
GE 160.0 252100
GEC 25.6 336600
GEF 125.49 376400
GEO 45.6 2618900
GES 81.82 629100
GET 55.07 331400
GEX 31.48 1467700
GF 11.56 261800
GFA 37.17 370400
GFF 23.92 4692100
GFI 21.58 1239200
GG 44.16 11267200
GGB 51.05 905400
GGG 45.69 330700
GHI 8.0 740700
GHL 81.0 324600
GIB 45.25 1606000
GIL 60.65 272000
GIM 9.98 3038400
GIS 83.25 520200
GKK 29.0 2464500
GLD 90.96 270400
GLF 64.58 619700
GLG 11.26 341900
GLS 17.34 360500
GLT 14.49 2418500
GLW 253.0 1348600
GME 58.41 488200
GMR 47.74 366800
GMT 62.81 496000
GMXR 47.48 254500
GNA 14.94 271500
GNK 83.37 361800
GNV 1.48 13621500
GNW 35.15 2671800
GOL 33.86 365300
GOM 23.1 291400
GOV 23.8 255500
```

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.

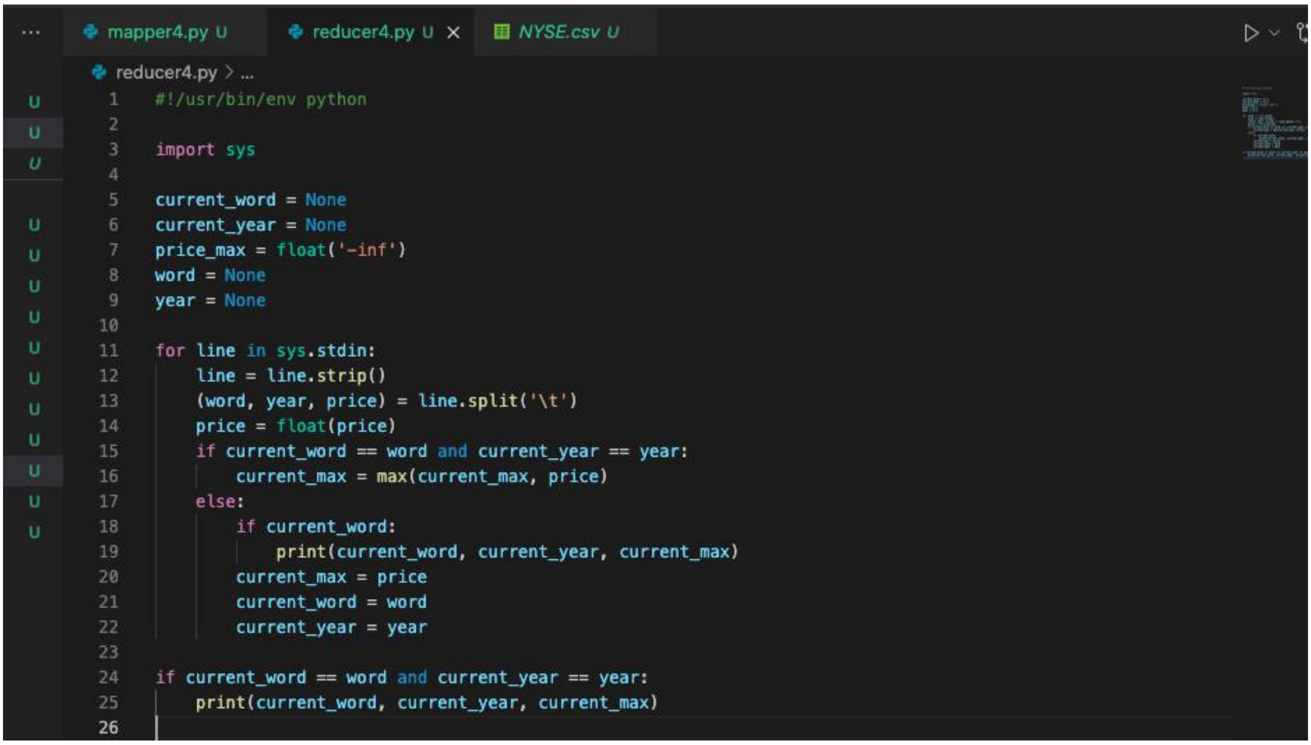
For example: output key and value of reduce, GYC 2019 18.15, GYC 2018 20.6, GYB 2020 21.41, GY 2019 8.43, GY 2018 8.64...

2-



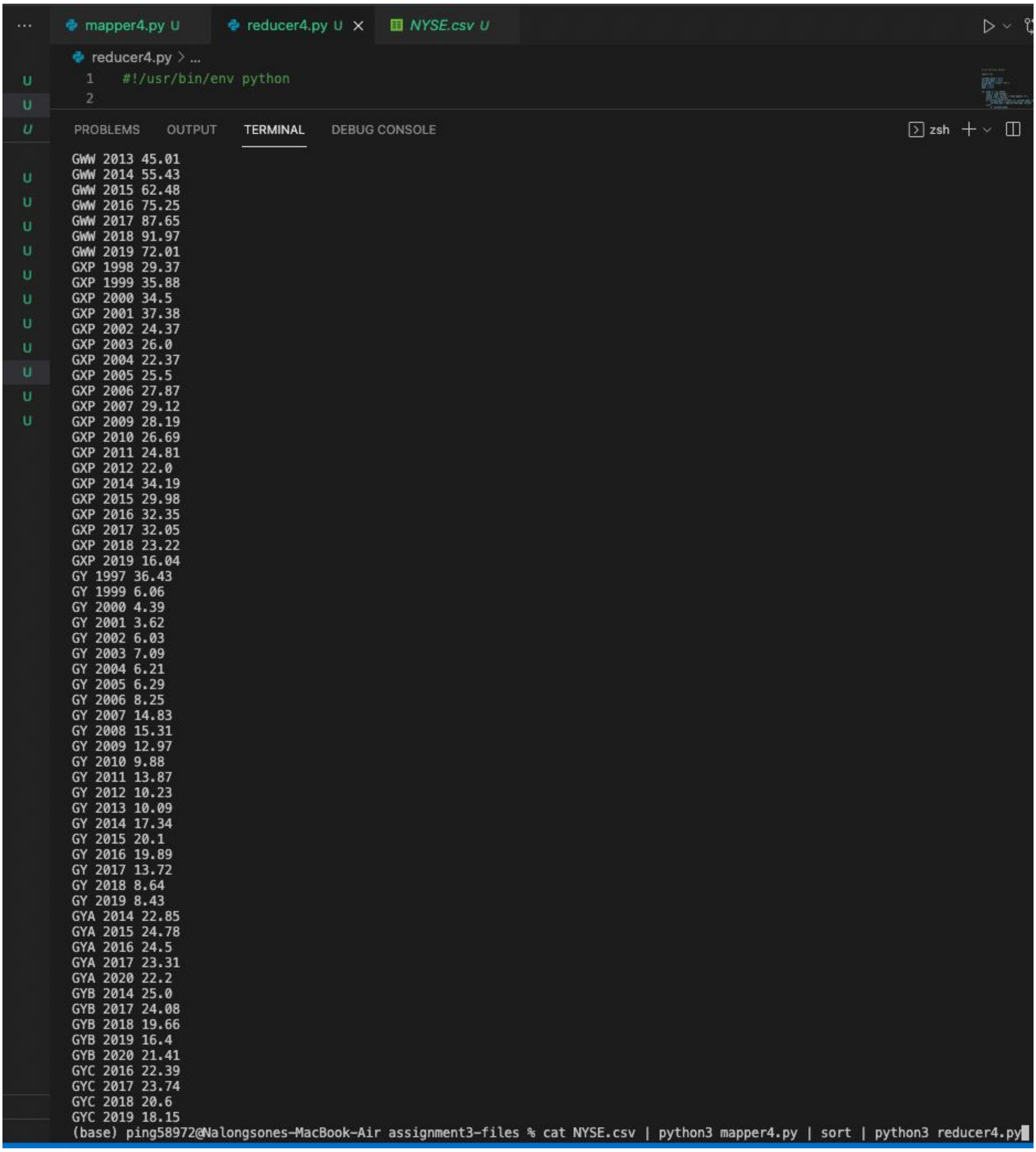
```python
#!/usr/bin/env python

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-



```python
#!/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-



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

```

```
GWW 2013 45.01
GWW 2014 55.43
GWW 2015 62.48
GWW 2016 75.25
GWW 2017 87.65
GWW 2018 91.97
GWW 2019 72.01
GXP 1998 29.37
GXP 1999 35.88
GXP 2000 34.5
GXP 2001 37.38
GXP 2002 24.37
GXP 2003 26.0
GXP 2004 22.37
GXP 2005 25.5
GXP 2006 27.87
GXP 2007 29.12
GXP 2009 28.19
GXP 2010 26.69
GXP 2011 24.81
GXP 2012 22.0
GXP 2014 34.19
GXP 2015 29.98
GXP 2016 32.35
GXP 2017 32.05
GXP 2018 23.22
GXP 2019 16.04
GY 1997 36.43
GY 1999 6.06
GY 2000 4.39
GY 2001 3.62
GY 2002 6.03
GY 2003 7.09
GY 2004 6.21
GY 2005 6.29
GY 2006 8.25
GY 2007 14.83
GY 2008 15.31
GY 2009 12.97
GY 2010 9.88
GY 2011 13.87
GY 2012 10.23
GY 2013 10.09
GY 2014 17.34
GY 2015 20.1
GY 2016 19.89
GY 2017 13.72
GY 2018 8.64
GY 2019 8.43
GYA 2014 22.85
GYA 2015 24.78
GYA 2016 24.5
GYA 2017 23.31
GYA 2020 22.2
GYB 2014 25.0
GYB 2017 24.08
GYB 2018 19.66
GYB 2019 16.4
GYB 2020 21.41
GYC 2016 22.39
GYC 2017 23.74
GYC 2018 20.6
GYC 2019 18.15
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat NYSE.csv | python3 mapper4.py | sort | python3 reducer4.py
```

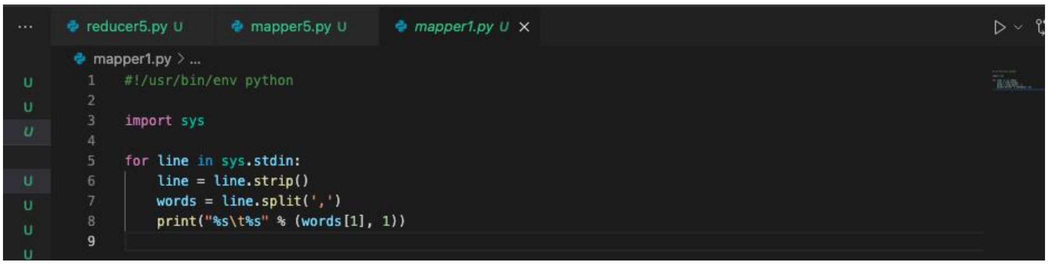
Task 1.5:

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

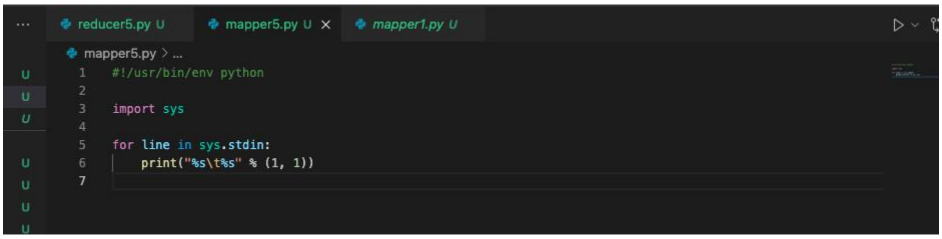
2-

mapper1.py



```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))

```

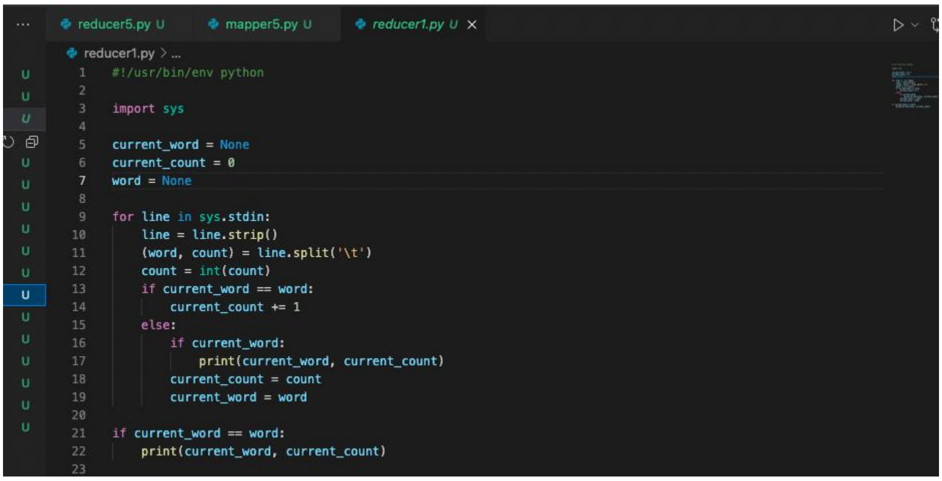
mapper5.py



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

import sys

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

```

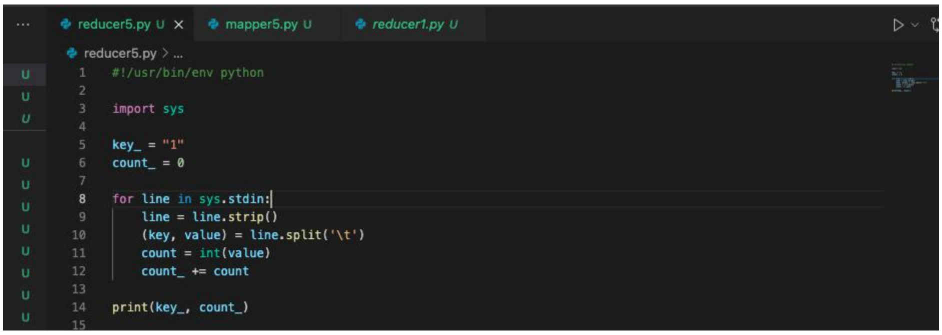
3-

reducer1.py



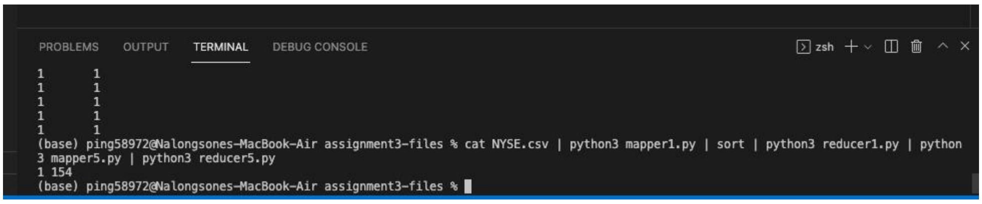
```python
#!/usr/bin/env python

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-



```
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

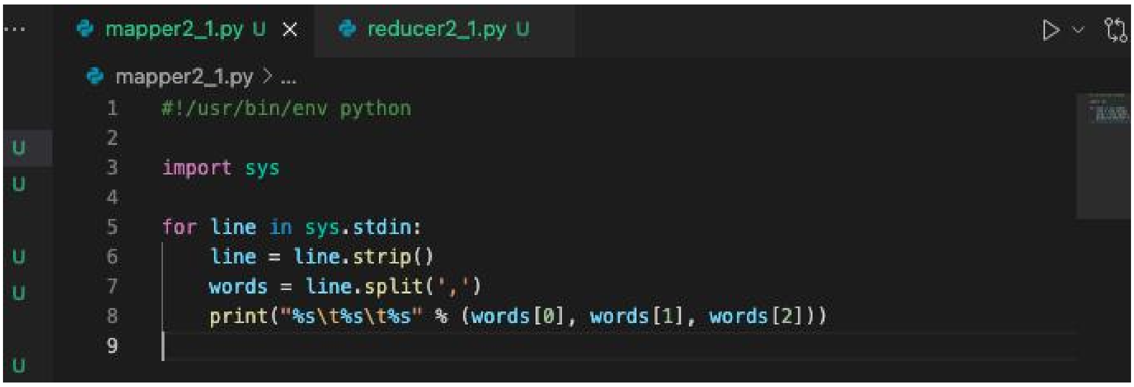
1002 Harness 150

1003 Full Charger 125

1004 Big Helmet 40

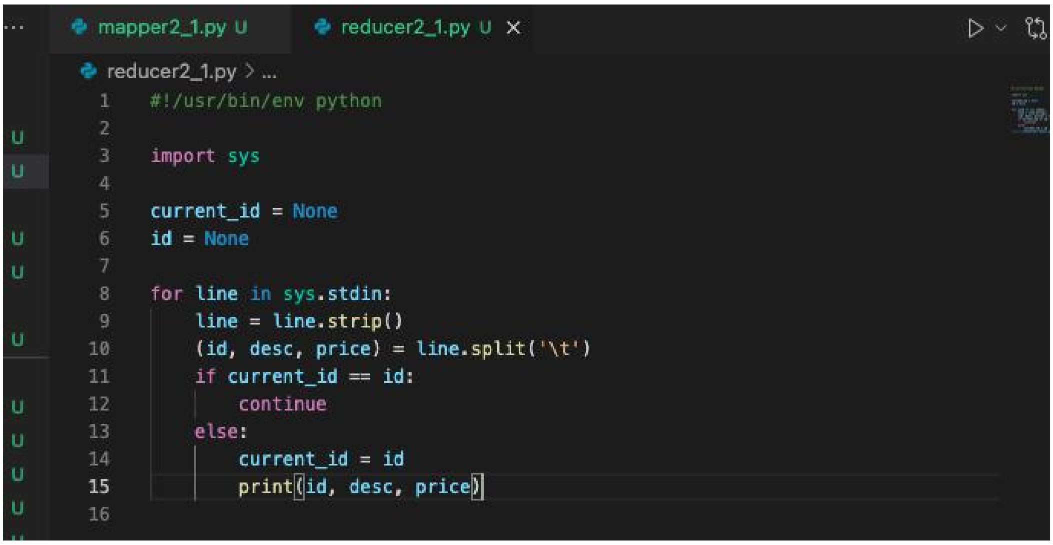
1009 Small Helmet 40

```
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-
```



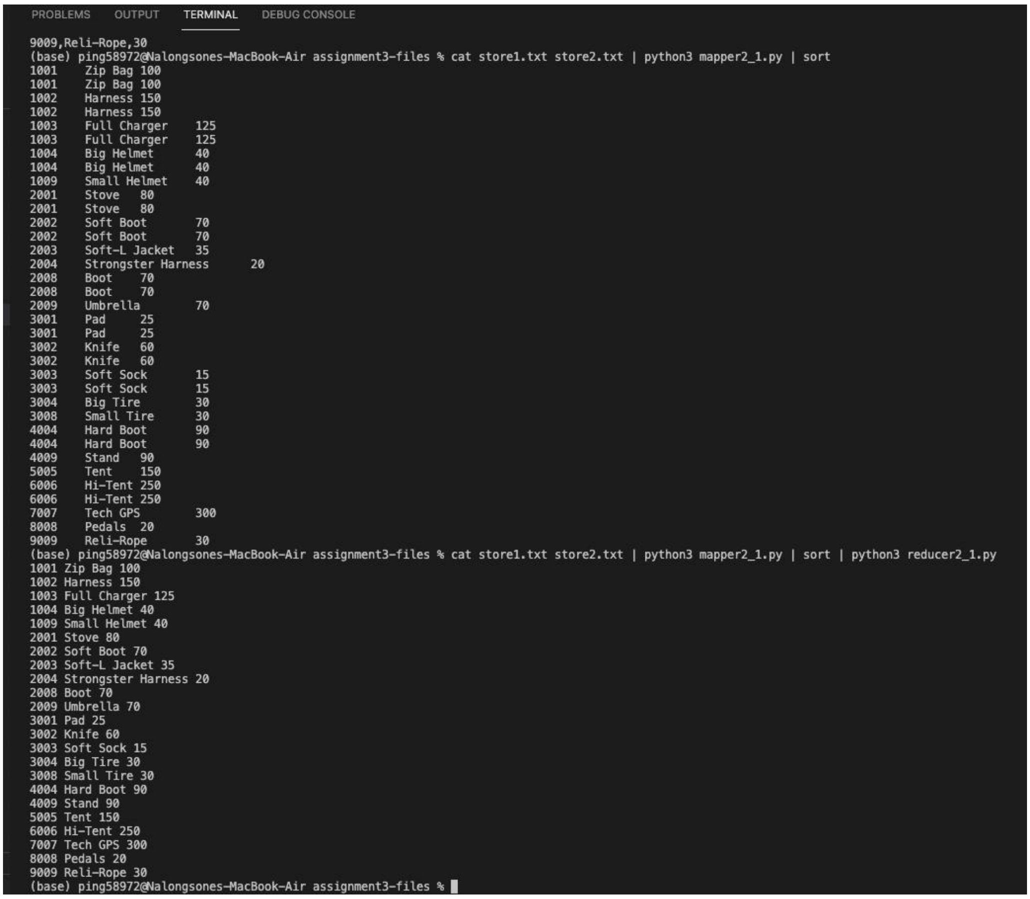
```python
#!/usr/bin/env python

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-



```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:
        continue
    else:
        current_id = id
        print(id, desc, price)

```

4-



```
9009,Reli-Rope,30
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % cat store1.txt store2.txt | python3 mapper2_1.py | sort
1001    Zip Bag 100
1001    Zip Bag 100
1002    Harness 150
1002    Harness 150
1003    Full Charger    125
1003    Full Charger    125
1004    Big Helmet      40
1004    Big Helmet      40
1009    Small Helmet    40
2001    Stove   80
2001    Stove   80
2002    Soft Boot       70
2002    Soft Boot       70
2003    Soft-L Jacket   35
2004    Strongster Harness      20
2008    Boot    70
2008    Boot    70
2009    Umbrella        70
3001    Pad     25
3001    Pad     25
3002    Knife   60
3002    Knife   60
3003    Soft Sock       15
3003    Soft Sock       15
3004    Big Tire        30
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_1.py | sort | python3 reducer2_1.py
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 30
(base) ping58972@Nalongsones-MacBook-Air assignment3-files % 
```

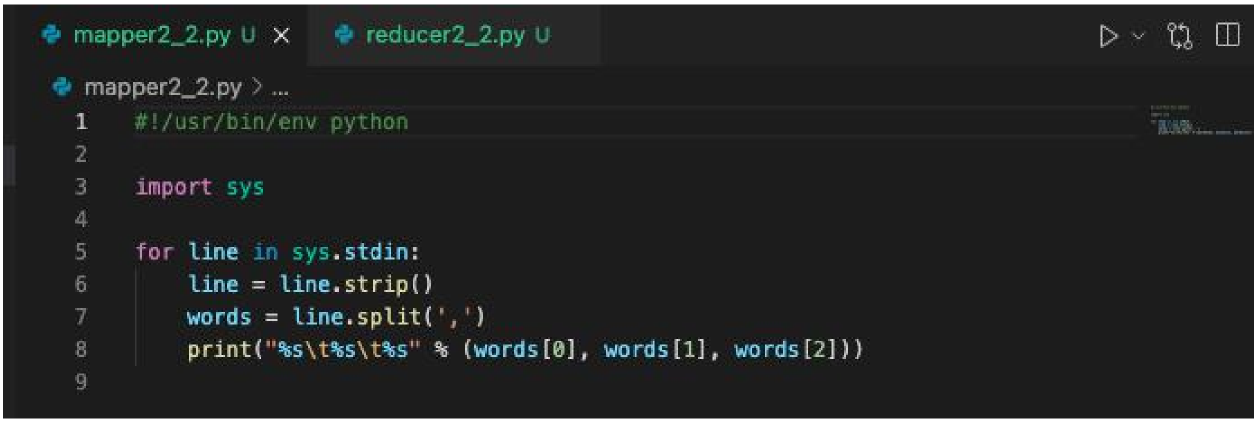
Task 2.2:

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:

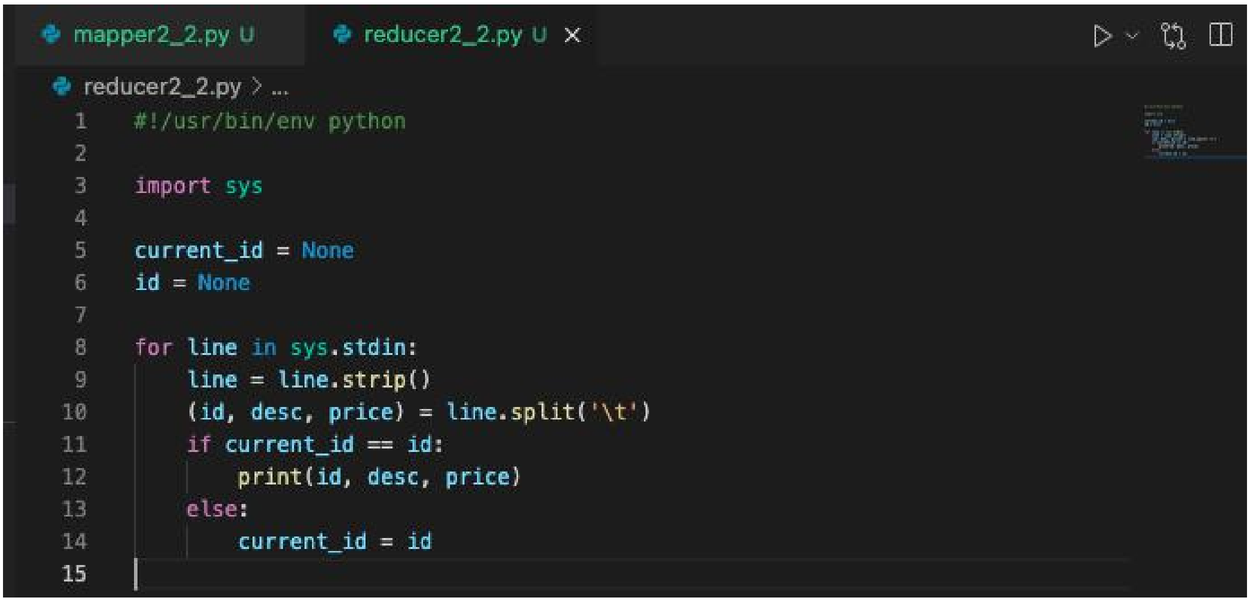
1001 Zip Bag 100
1002 Harness 150
1003 Full Charger 125
1004 Big Helmet 40
2001 Stove 80
2002 Soft Boot 70
2008 Boot 70
3001 Pad 25
3002 Knife 60
3003 Soft Sock 15
4004 Hard Boot 90
6006 Hi-Tent 250

2-



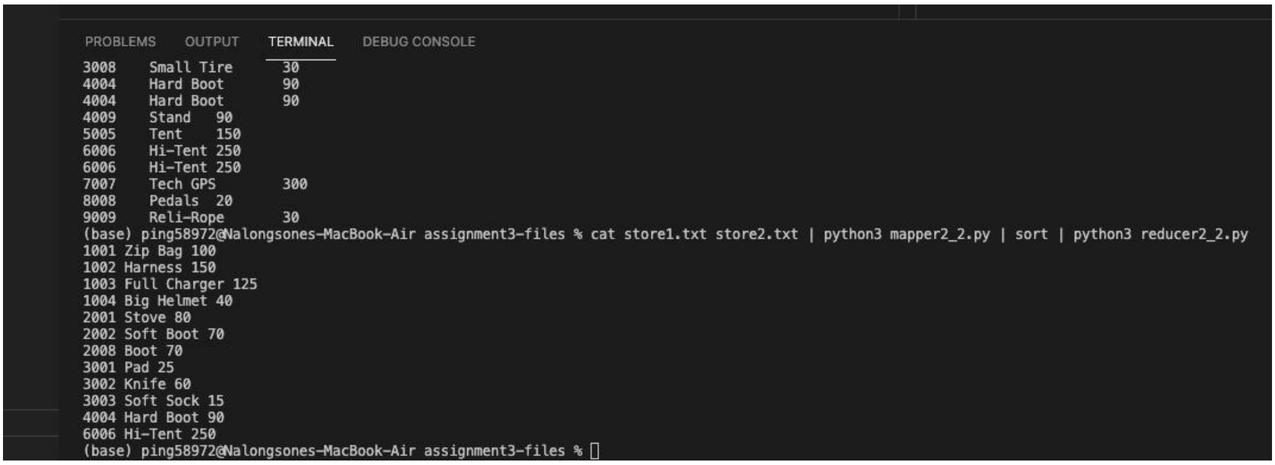
```python
#!/usr/bin/env python

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-



```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-



```
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 % ▯
```