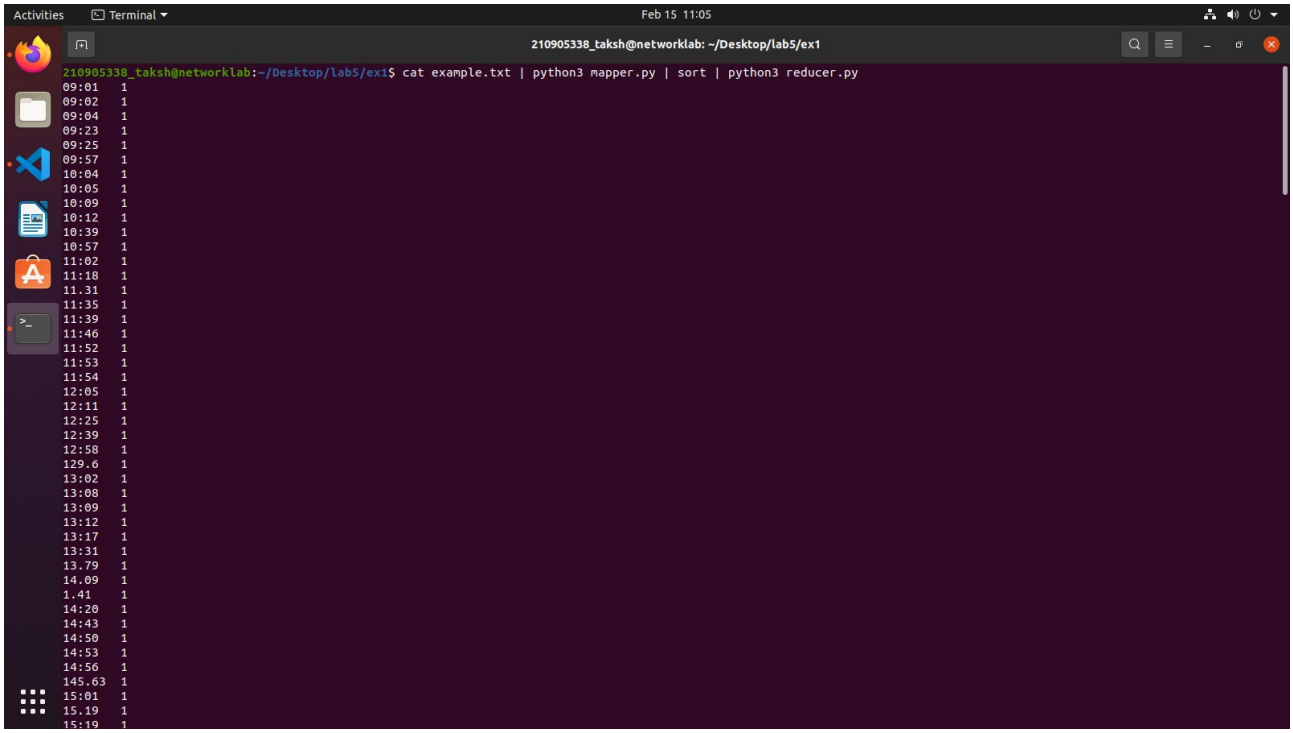
Q1. Word Count Program

mapper.py

```
import sys
for line in sys.stdin: # input comes from STDIN (standard input)
line = line.strip() # remove leading and trailing whitespace
words = line.split() # split the line into words
# increase counters
for word in words:
# write the results to STDOUT (standard output);
# what we output here will be the input for the
# Reduce step, i.e. the input for reducer.py #
# tab-delimited; the trivial word count is 1
print("%s\t\t%s" %(word, 1))
```

reducer.py

```
from operator import itemgetter
import sys
current_word = None
current_count = 0
word = None
# input comes from STDIN
for line in sys.stdin:
# remove leading and trailing whitespace
line = line.strip()
# parse the input we got from mapper.py
word, count = line.split('\t', 1)
# convert count (currently a string) to int
try:
count = int(count)
except ValueError:
# count was not a number, so silently
# ignore/discard this line
continue
# this IF-switch only works because Hadoop sorts map output
# by key (here: word) before it is passed to the reducer
if current_word == word:
current_count += count
else:
if current_word:
# write result to STDOUT
print('%s\t%s' % (current_word, current_count) )
current_count = count
current_word = word
# do not forget to output the last word if needed!
if current_word == word:
print('%s\t%s' % (current_word, current_count))
```



Q2. MapReduce to find frequent words

freqmap1.py

```
from __future__ import print_function
import sys
import pandas as pd
df = pd.read_excel('german.xlsx')
for _, row in df.iterrows():
line = ", ".join([str(value) for value in row.values])
words = line.split(',')
for word in words:
print( '%s\t%d' % (word, 1) )
```

freqred1.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
```

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

freqmap2.py

```
from __future__ import print_function
import sys
# input comes from STDIN (standard input)
for line in sys.stdin:
# print(line.strip().split())
word, count = line.strip().split('\t')
count = int(count)
print( '%d,%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(',', 1)
count = int(count)
if count > currentMax:
currentMax = count
mostFreq = [ word ]
elif count == currentMax:
mostFreq.append( word )
# output mostFreq word(s)
for word in mostFreq:
print( '%s, %s' % ( word, currentMax ) )
```

Q3. Count and Summary using MapReduce

itemmap.py

```
import sys
for line in sys.stdin:
data = line.strip().split(",")
if len(data) == 14:
age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal,target = data
print ("{0}\t{1}".format(age, target))
```

itemred.py

```
import sys
transactions_count = 0
age_total = 0
for line in sys.stdin:
data = line.strip().split("\t")
if len(data) != 2:
```

```
# Something has gone wrong. Skip this line.
continue
age, target = data
transactions_count += 1
if age.isdigit():
age_total += float(age)
print (transactions_count, "\t", age_total)
```