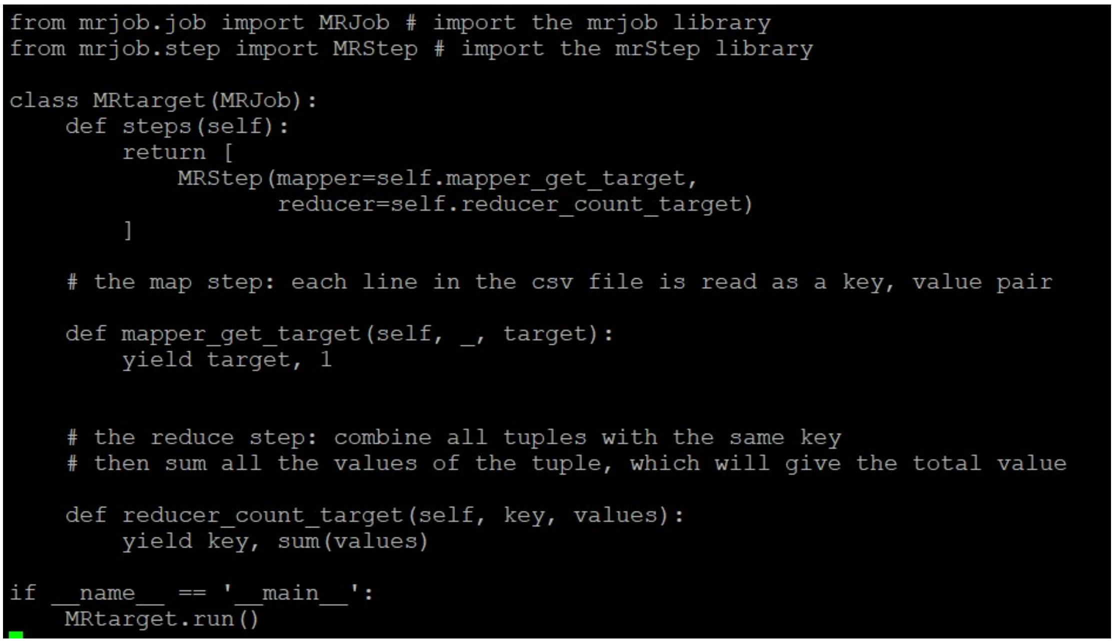
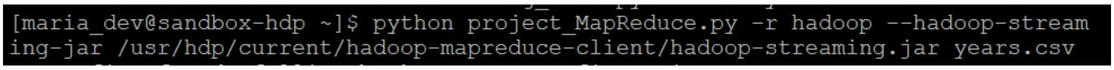
count of years to find out the year with the most shares

- MapReduce function:



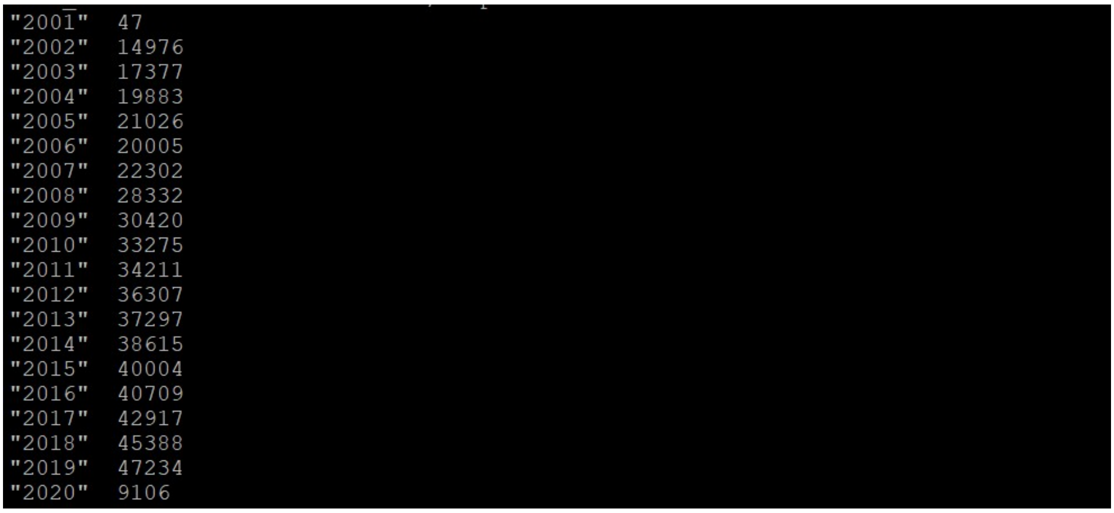
```
from mrjob.job import MRJob # import the mrjob library
from mrjob.step import MRStep # import the mrStep library

class MRtarget(MRJob):
    def steps(self):
        return [
            MRStep(mapper=self.mapper_get_target,
                   reducer=self.reducer_count_target)
        ]

    # the map step: each line in the csv file is read as a key, value pair

    def mapper_get_target(self, _, target):
        yield target, 1


    # the reduce step: combine all tuples with the same key
    # then sum all the values of the tuple, which will give the total value

    def reducer_count_target(self, key, values):
        yield key, sum(values)

if __name__ == '__main__':
    MRtarget.run()
```

- Running Hadoop Cluster on a years data:



```
[maria_dev@sandbox-hdp ~]$ python project_MapReduce.py -r hadoop --hadoop-stream
ing-jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar years.csv
```

- Output:
- 2019 is the most year in which shares were offered for trading



```
"2001"  47
"2002"  14976
"2003"  17377
"2004"  19883
"2005"  21026
"2006"  20005
"2007"  22302
"2008"  28332
"2009"  30420
"2010"  33275
"2011"  34211
"2012"  36307
"2013"  37297
"2014"  38615
"2015"  40004
"2016"  40709
"2017"  42917
"2018"  45388
"2019"  47234
"2020"  9106
```