```
sjcet@ADLAB:~$ cat data.txt
DEAR BEAR RIVER CAR CAR RIVER DEAR CAR BEAR
sjcet@ADLAB:~$ cat mapper.py
#!/usr/bin/env python

# import sys because we need to read and write data to STDIN and STDOUT
import sys

# reading entire line from STDIN (standard input)
for line in sys.stdin:
        # to remove leading and trailing whitespace
        line = line.strip()
        # split the line into words
        words = line.split()

        # we are looping over the words array and printing the word
        # with the count of 1 to the STDOUT
        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
                print('%s\t%s' % (word, 1))
```

```
sjcet@ADLAB:~$ cat reducer.py
#!/usr/bin/env python

from operator import itemgetter
import sys

current_word = None
current_count = 0
word = None

# read the entire line from STDIN
for line in sys.stdin:
        # remove leading and trailing whitespace
        line = line.strip()
        # splitting the data on the basis of tab we have provided in 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))
sjcet@ADLAB:~$
```

```
sjcet@ADLAB:~$ cat data.txt | python mapper.py
DEAR    1
BEAR    1
RIVER   1
CAR     1
CAR     1
RIVER   1
DEAR    1
CAR     1
BEAR    1
sjcet@ADLAB:~$ cat data.txt | python mapper.py | sort -k1,1 | python reducer.py
BEAR    2
CAR     3
DEAR    2
RIVER   2
sjcet@ADLAB:~$
```