

```
In [ ]: from mrjob.job import MRJob

class WordCount(MRJob):

    def mapper(self, _, line):
        words = line.split()
        for word in words:
            if word.isalpha() and len(word) > 0:
                yield word[0].lower(), 1

    def reducer(self, letter, counts):
        yield letter, sum(counts)

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

```
In [ ]: class CountWordsStartingwithEachAlphabet(MRJob):

    def mapper(self, _, line):
        yield "chars", len(line)
        yield "words", len(line.split())
        yield "lines", 1

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

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

```
In [ ]: class WordCountLength(MRJob):

    def mapper(self, _, line):
        words = line.split()
        for word in words:
            if len(word) == 5:
                yield word, 1

    def reducer(self, word, counts):
        yield word, sum(counts)

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