

These questions are based on the Udacity course “Intro to Hadoop and MapReduce”.

1. Provide your solution to making the Mapper code more defensive:

```
1 #!/usr/local/bin/python
2
3 import sys
4
5 # read standard input line by line
6 for line in sys.stdin:
7     # strip off extra whitespace, split on tab
8     # and put the data in an array
9     data = line.strip().split("\t")
10
11    # This is the place you need to do some defensive programming
12    # what if there are not exactly 6 fields in that line?
13    # YOUR CODE HERE
14
15
16
17
18    # this next line is called 'multiple assignment' in Python
19    # this is not really necessary, we could access the data
20    # with data[2] and data[5], but we do this for conveniency
21    # and to make the code easier to read
22    date, time, store, item, cost, payment = data
23
24    # Now print out the data that will be passed to the reducer
25    print "{0}\t{1}".format(store, cost)
```

2. Complete the matching Reducer code:

```
1 #!/usr/local/bin/python
2
3 import sys
4
5 salesTotal = 0
6 oldKey = None
7
8 for line in sys.stdin:
9     data = line.strip().split("\t")
10    if len(data) != 2:
11        continue
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
```

To test your solution, download the file `purchases.tgz` from GitHub<sup>1</sup>, unpack it and copy the file `purchases.txt` into the miniHive Docker container. Then run the following command inside the miniHive Docker container:

```
cat purchases.txt | python mapper.py | sort | python reducer.py
```

---

<sup>1</sup><https://github.com/sdbs-uni-p/sds-artifacts/tree/main>