

In [2]:

```
import sys
import string
import logging
from util import mapper_logfile #provide util file
logging.basicConfig(filename=mapper_logfile, format='%(message)s',
                    level=logging.INFO, filemode='w')

def mapper():
    """
    The input to this mapper will be the final Subway-MTA dataset, the same as
    in the previous exercise.  You can check out the csv and its structure below:
    https://s3.amazonaws.com/content.udacity-data.com/courses/ud359/turnstile_data_master_with_weather.csv

    For each line of input, the mapper output should PRINT (not return) the UNIT as
    the key, the number of ENTRIESn_hourly as the value, and separate the key and
    the value by a tab. For example: 'R002\t105105.0'

    Since you are printing the output of your program, printing a debug
    statement will interfere with the operation of the grader. Instead,
    use the logging module, which we've configured to log to a file printed
    when you click "Test Run". For example:
    logging.info("My debugging message")
    Note that, unlike print, logging.info will take only a single argument.
    So logging.info("my message") will work, but logging.info("my","message") will not.

    The logging module can be used to give you more control over your debugging
    or other messages than you can get by printing them. In this exercise, print
    statements from your mapper will go to your reducer, and print statements
    from your reducer will be considered your final output. By contrast, messages
    logged via the loggers we configured will be saved to two files, one
    for the mapper and one for the reducer. If you click "Test Run", then we
    will show the contents of those files once your program has finished running.
    The logging module also has other capabilities; see
    https://docs.python.org/2/library/logging.html for more information.
    """

    for line in sys.stdin:
        data = line.strip().split(',')
        if data[1] != 'UNIT' and len(data) == 22:
            print ('{0}\t{1}'.format(data[1], data[6]))
mapper()
```

In []:

In []: