1. Problem Set 5.1
3. # mapper.py
5. **import** sys
6. **import** string
7. **import** logging
9. **from** util **import** mapper\_logfile
10. logging.basicConfig(filename=mapper\_logfile, format='%(message)s',
11. level=logging.INFO, filemode='w')
13. **def** mapper():
15. **for** line **in** sys.stdin:
16. # your code here
17. data = line.strip().split(',')
18. **if** len(data) != 22 **or** data[1] == 'UNIT':
19. **continue**
21. out = "{}\t{}".format(data[1], data[6])
22. **print** out

25. mapper()
27. **import** sys
28. **import** logging
30. **from** util **import** reducer\_logfile
31. logging.basicConfig(filename=reducer\_logfile, format='%(message)s',
32. level=logging.INFO, filemode='w')
34. **def** reducer():
36. entries = 0
37. last\_unit = None
39. **for** line **in** sys.stdin:
40. # your code here
41. data = line.strip().split('\t')
42. **if** len(data) != 2:
43. **continue**
44. this\_unit, count = data
46. **if** last\_unit **and** last\_unit != this\_unit:
47. **print** '{0}\t{1}'.format(last\_unit, entries)
48. entries = 0
50. last\_unit = this\_unit
51. entries += float(count)
53. **if** last\_unit != None:
54. **print** "{}\t{}".format(last\_unit, entries)

57. reducer()



62. Problem Set 5.2
64. **import** sys
65. **import** string
66. **import** logging
68. **from** util **import** mapper\_logfile
69. logging.basicConfig(filename=mapper\_logfile, format='%(message)s',
70. level=logging.INFO, filemode='w')
72. **def** mapper():
73. **def** format\_key(fog, rain):
74. **return** '{}fog-{}rain'.format(
75. '' **if** fog **else** 'no',
76. '' **if** rain **else** 'no'
77. )
79. **for** line **in** sys.stdin:
80. # your code here
81. data = line.strip().split(',')
82. **if** len(data) != 22 **or** data[1] == 'UNIT':
83. **continue**
84. out = "{}\t{}".format(format\_key(float(data[14]), float(data[15])), data[6])
85. **print** out


89. mapper()
90. **import** sys
91. **import** logging
93. **from** util **import** reducer\_logfile
94. logging.basicConfig(filename=reducer\_logfile, format='%(message)s',
95. level=logging.INFO, filemode='w')
97. **def** reducer():
99. riders = 0      # The number of total riders for this key
100. num\_hours = 0   # The number of hours with this key
101. old\_key = None
103. **for** line **in** sys.stdin:
104. # your code here
105. data = line.strip().split('\t')
106. **if** len(data) != 2:
107. **continue**
108. this\_unit, count = data
110. **if** old\_key **and** old\_key != this\_unit:
111. **print** '{0}\t{1}'.format(old\_key, (riders/float(num\_hours)))
112. riders = 0
113. num\_hours = 0
115. old\_key = this\_unit
116. riders += float(count)
117. num\_hours += 1
119. **if** old\_key != None:
120. **print** '{0}\t{1}'.format(old\_key, (riders/float(num\_hours)))
121. reducer()
122. Problem Set 5.3
124. **import** sys
125. **import** string
126. **import** logging
128. **from** util **import** mapper\_logfile
129. logging.basicConfig(filename=mapper\_logfile, format='%(message)s',
130. level=logging.INFO, filemode='w')
132. **def** mapper():
133. **for** line **in** sys.stdin:
134. data = line.strip().split(',')
135. **if** data[1] == 'UNIT' **or** len(data) != 22:
136. **continue**
138. **print** '{0}\t{1}\t{2}\t{3}'.format(data[1],data[6], data[2], data[3])

141. mapper()
143. **import** sys
144. **import** logging
146. **from** util **import** reducer\_logfile
147. logging.basicConfig(filename=reducer\_logfile, format='%(message)s',
148. level=logging.INFO, filemode='w')
149. **def** reducer():
150. max\_entries = 0
151. old\_key = None
152. datetime = ''
154. **for** line **in** sys.stdin:
155. # your code here
156. data = line.strip().split('\t')
158. **if** len(data) != 4:
159. **continue**
161. this\_key, count, date, time  = data
162. count = float(count)
164. **if** old\_key **and** old\_key != this\_key:
165. **print** "{0}\t{1}\t{2}".format(old\_key, datetime, max\_entries)
166. max\_entries = 0
167. datetime = ''
169. old\_key = this\_key
170. **if** count >= max\_entries:
171. max\_entries = count
172. datetime = str(date) + ' ' + str(time)
173. #datetime = datetime.strptime(date, '%Y-%m-%d') + ' ' + datetime.strptime(time, '%H:%M:%S')
175. **if** old\_key != None:
176. **print** "{0}\t{1}\t{2}".format(old\_key, datetime, max\_entries)
178. reducer()