[Project] 서울 공기질 분석

문제 1) 각 지역(station code)별로 평균, 최대, 최소 PM10 측정치 구하기

각 지역별로 평균, 최대, 최소 PM10을 구하기 위하여 map 함수에서 output key를 station code, output value를 average value 값으로 주었습니다. 입력 csv를 한 줄씩 읽으면서 파일의 경우 첫 행이 열 정보를 담고 있기 때문에 넘어가주었고, item code가 8이 아니면 넘어가주었습니다. 또한, instrument status가 0인 경우가 기기가 정상인 경우이므로 이외의 경우는 넘어가주었습니다. 모든 조건을 만족한 경우에만 station code와 average value를 emit해주었습니다.

reduce 함수에서는 입력으로 들어온 station code별 average value에 대하여 각각 평균, 최소, 최대값을 구하여 emit해주었습니다.

```
101	Average: 37.733, Min: 3.000, Max: 289.000
104	Average: 42.104, Min: 1.000, Max: 423.000
107	Average: 44.344, Min: 3.000, Max: 411.000
110	Average: 39.209, Min: 3.000, Max: 414.000
113	Average: 40.849, Min: 1.000, Max: 354.000
116	Average: 43.938, Min: 3.000, Max: 389.000
119	Average: 47.120, Min: 3.000, Max: 351.000
122	Average: 44.460, Min: 1.000, Max: 470.000
125	Average: 45.148, Min: 1.000, Max: 443.000
```

[part-r-00000]

```
102	Average: 38.043, Min: 3.000, Max: 296.000
105	Average: 42.612, Min: 3.000, Max: 401.000
108	Average: 41.404, Min: 3.000, Max: 340.000
111	Average: 44.383, Min: 3.000, Max: 421.000
114	Average: 40.534, Min: 3.000, Max: 289.000
117	Average: 43.385, Min: 3.000, Max: 405.000
120	Average: 41.863, Min: 3.000, Max: 321.000
123	Average: 39.868, Min: 1.000, Max: 302.000
```

[part-r-00001]

```
103	Average: 35.903, Min: 3.000, Max: 330.000
106	Average: 43.930, Min: 3.000, Max: 389.000
109	Average: 39.775, Min: 3.000, Max: 326.000
112	Average: 39.003, Min: 2.000, Max: 322.000
115	Average: 42.795, Min: 3.000, Max: 293.000
118	Average: 39.659, Min: 3.000, Max: 329.000
121	Average: 44.953, Min: 3.000, Max: 385.000
124	Average: 42.441, Min: 1.000, Max: 426.000
```

[part-r-00002]

문제 2) PM10, PM2.5 기준으로 공기의 질이 '좋음' 수준이 가장 많이 측정된 지역은 어디인지 찾기
(공기질 좋음 = PM10 기준: 30 이하, PM2.5 기준 15 이하)

map 함수에서는 reduce 함수에서 지역마다의 공기의 질 '좋음'이 나타난 횟수를 셀 수 있도록 하기 위해서 output key는 station code, output value는 1로 할당해주었다. 1번에서와 마찬가지로 입력 첫 줄은 무시해주고, item code가 8, 9가 아닌 경우, instrument status가 0이 아닌 경우를 무시해주었다. 또한, item code가 8이고 average value가 30보다 큰 경우, item code가 9이고 average value가 15보다 큰 경우도 무시해주었다. 이외의 경우에 대하여 앞서 말한 output key, value를 emit해주었다.

reduce 함수에서는 입력으로 들어온 station code에 대응하는 모든 1 값들을 합해 개수를 구해주고, key를 station code, value를 1의 개수로 하여 emit해주었다.

이후 별도로 공기의 질이 '좋음' 수준이 가장 많이 측정된 지역을 구하기 위하여 get_good_station()이라는 함수를 구현하여 그 지역을 출력해주었다.

```
good air pollution station is 112
```

[출력 결과]

101	21876	102	22036	103	22498
104	18920	105	19608	106	16971
107	18705	108	19487	109	20553
110	21082	111	18425	112	22993
113	21487	114	19253	115	18200
116	18513	117	19559	118	19730
119	16113	120	18960	121	15946
122	19652	123	20903	124	19848
125	17955				

[part-r-00000] [part-r-00001] [part-r-00002]

문제 3) 데이터 변환하기 → 각 〈시간, 지역〉별로 모든 종류의 측정치 모아서 저장하기

map 함수에서는 〈시간, 지역〉 별로 측정 치를 모으기 위하여 output key를 Text(measurement date, average value), ouput value를 Text(item code, average value)로 구현하였다. 위와 마찬가지로 입력의 첫 줄은 무시하고, StringTokenizer로 토큰화 한 값들을 각 변수에 저장해둔 후, instrument status가 0인 경우 key와 value 값으로 재구성하여 emit 해주었다.

reduce 함수에서는 value로 들어온 값들을 item code 오름차순으로 정렬될 수 있도록 하기 위하여, 다시 토큰화 한 후 average value들을 item code 순서(1, 3, 5, 6, 8, 9)로 재구성하여 output value로 지정해주었다. Output key는 입력으로 들어온 키 값 그대로 Text(measurement date, average value)로 하여 emit해주었다.

```
2017-01-01 00:00,101	0.004 0.059000000000000004 1.2 0.002 73.0 57.0
2017-01-01 00:00,104	0.005 0.045 0.6 0.003 73.0 46.0
2017-01-01 00:00,107	0.005 0.049 0.9 0.002 64.0 40.0
2017-01-01 00:00,110	0.005 0.04 0.8 0.002 91.0 50.0
2017-01-01 00:00,113	0.006 0.051 0.9 0.002 81.0 40.0
2017-01-01 00:00,116	0.006999999999999999 0.07 1.3 0.002 107.0 65.0
2017-01-01 00:00,119	0.005 0.035 1.5 0.004 70.0 46.0
2017-01-01 00:00,122	0.005 0.039 1.3 0.005 82.0 39.0
2017-01-01 00:00,125	0.004 0.042 0.9 0.002 68.0 53.0
2017-01-01 01:00,103	0.004 0.038 1.4 0.002 73.0 66.0
2017-01-01 01:00,106	0.004 0.064 1.5 0.003 70.0 62.0
2017-01-01 01:00,109	0.005 0.05 1.2 0.002 62.0 49.0
2017-01-01 01:00,112	0.005 0.046 1.0 0.002 60.0 52.0
2017-01-01 01:00,115	0.005 0.053 1.3 0.002 74.0 49.0
2017-01-01 01:00,118	0.004 0.059000000000000004 1.3 0.001 66.0 45.0
2017-01-01 01:00,121	0.006 0.075 1.5 0.004 70.0 51.0
2017-01-01 01:00,124	0.005 0.036000000000000004 1.3 0.003 65.0 43.0
```

[part-r-00000의 일부]

문제 4) 시간대를 기준으로 평균 공기질 구하기 (SO2, NO2, CO, O3, PM10, PM2.5 한꺼번에 구하기)

map 함수에서는 문제 3과 동일하게 진행하지만, key값을 time으로 하기 위하여 StringTokenizer를 이용하여 Measurement date에서 time만을 분리해냈다. 따라서 output key를 Text(time), output value를 Text(item code, average value)로 하여 emit해주었다.

reduce 함수에서는 value들을 다시 분리하여 item code별로 모아 평균을 구해주었다.

```
02:00  0.004124 0.025840 0.528401 0.019201 39.853836 24.104321
05:00  0.004044 0.027084 0.526526 0.015178 38.270653 23.312805
08:00  0.004382 0.033379 0.588348 0.014962 42.249924 24.184484
11:00  0.004563 0.026756 0.503218 0.029342 43.945175 24.291567
14:00  0.004444 0.022554 0.446770 0.040878 43.550022 23.754595
17:00  0.004400 0.027084 0.461889 0.035169 42.831188 23.535530
20:00  0.004360 0.033299 0.531068 0.022653 43.164486 25.227777
23:00  0.004245 0.032782 0.542516 0.018129 40.834991 24.406862

01:00  0.004162 0.027941 0.536454 0.018936 40.189247 24.040787
04:00  0.004051 0.024580 0.518077 0.017839 38.663952 23.626789
07:00  0.004222 0.033590 0.574503 0.012328 39.890850 23.418598
10:00  0.004574 0.029252 0.537472 0.023924 44.573174 24.765680
13:00  0.004473 0.023043 0.459027 0.038606 42.953014 23.685593
16:00  0.004428 0.024398 0.447408 0.039357 43.627678 23.813921
19:00  0.004372 0.032740 0.516815 0.025594 42.666885 24.115837
22:00  0.004293 0.033403 0.541211 0.018994 41.799412 24.991611

00:00  0.004198 0.030758 0.541853 0.018235 41.089653 24.470604
03:00  0.004083 0.024584 0.521074 0.018911 38.598225 23.494148
06:00  0.004089 0.031196 0.548106 0.012429 39.250732 23.637861
09:00  0.004522 0.031553 0.571594 0.019093 43.253498 24.246250
12:00  0.004521 0.024349 0.477248 0.034719 43.734142 24.276747
15:00  0.004434 0.022958 0.440705 0.041308 43.271736 23.465151
18:00  0.004386 0.030431 0.490490 0.029896 42.953369 23.943842
21:00  0.004332 0.033263 0.537134 0.020641 42.202888 25.143055
```