

### Part(a)

class Mapper

method map(docid a, doc d)

site = r.getSensorId()

temp = r.getTemperature()

Emit(site, temp)

#### Note:

1. There is no loop in this map method. Reason: Each record has exactly one sensorID and one temperature.
2. There is no need to even get time. Reason: Time is never used.
3. Map can emit only a key value pair.

class Reducer

method Reduce(string site, Int[ v1, v2, v3, v4, v5 ...]) // if you use double that is fine as well.

sum = v1, count = v1, max = v1, min = v1

for all v in [v2, v3, v4, v5 ...] do

sum += v, count++, max = Max(max, v) , min = Min(min, v)

threeVal = new ThreeValues(max, min, sum/count)

//ThreeValues is a custom class with two integers and a double

Emit(site, threeVal)

#### Note:

1. Reduce's signature must match Map emit's signature
2. Reduce can emit only a key value pair.  
I have site is the key. All other values, I packaged it in a class class "ThreeValues".

### Part (b)

class Mapper

method initialize()

H = new AssociativeArray()

```
method map(docid a, doc d)
```

```
    site = r.getSensorId()
```

```
    temp = r.getTemperature()
```

```
    if (H{site} is null)
```

```
        H{site} = new FourInt(temp, 1, temp, temp)
```

```
        //FourInt is a custom class with four integers
```

```
        // They are Sum, Count, Max and Min found so far
```

```
    else
```

```
        H{site}.Sum += temp, H{site}.Count++
```

```
        if (H{site}.Max < temp) H{site}.Max = temp
```

```
        else If (H{site}.Min > temp) H{site}.Min = temp
```

```
method close()
```

```
    for all term t in H do
```

```
        Emit(t, H{t})
```

```
        // Note that H{t} is a FourInt. Map can emit only a key and a value.
```

```
class Reducer
```

```
method Reduce(string site, FourInt [ v1, v2, v3, v4, v5 ...])
```

```
    sum = v1.Sum, count = v1.Count, max = v1.Max, min = v1.min
```

```
    for all FourInt v in [v2, v3, v4, v5 ...] do
```

```
        sum += v.Sum, count += v.Count, max = Max(max, v.Max) , min = Min(min, v.min)
```

```
    threeVal = new ThreeValues(max, min, sum/count)
```

```
        //ThreeValues is a custom class with two integers and a double
```

```
    Emit(site, threeVal)
```