Requirements:

Implement a Spark batch job to aggregate the data from the sensors by time slots and locations. Use **Java** and **Spark SQL API** to do the next tasks:

1. The main task is: Load the both input data sources (Sensor Devices and Sensor Data) as Dataframes and using the transformations and actions write the resulting Dataframe (Location Data Time Slots 15 minutes) into new JSON file.
    a. The Spark Job is parameterized with a from and until date that determines which time slots to calculate. This includes time slots **without data** in the Input Data Source 2 (e.g from=2018-09-01 until=2018-09-03 must generate Output Data Source 15 minutes file with 288 records for each Room (3 days * 24 hours * 4 time slots) and Output Data Source 1 hour file with 72 records for each Room (3 days * 24 hours) even there is no data for some time slots)
2. Additional task is: Load the output data source 1 (Location Data Time Slots 15 minutes) as RDD and using the transformations and actions write the resulting RDD (Location Data Time Slots 1 hour) into new JSON file.

Input Data Source 1:

CSV file with following structure:

- SensorId – String; unique ID of a Sensor
- CnanelId – String; unique within SensorId ("SensorId + ChannelId" is unique identifier of a record)
- ChannelType – String; can be on of the next values ["temperature", "battery", "presence"]
- LocationId – String; Id of a Room

Example:

SensorId	ChannelID	ChannelType	LocationId
id1	ch1	temperature	room2
id1	ch2	battery	room2
Id2	ch1	temperature	room1
Id2	ch2	battery	room1

Id2	ch3	presence	room1
Id3	ch1	presence	room2

Input Data Source 2:

CSV file with following structure:

- SensorId – String; unique ID of a Sensor
- CnanelId – String; ID of a Sensor's Channel
- Timestamp – String; ISO Format of Local Date Time(without offset); Date and Time when the Sensor send the Value ("SensorId + ChannelId + TimeStamp" is unique identifier of a record)
- Value – String; Value sent by the Sensor for a particular Channel; possible values depends on the ChannelType:
  - temperature – value in Fahrenheit (e.g "70.3", "72", "73.234");
  - battery – value in range from 0 to 100 stored as String; relative battery health
  - presence – can be "0" or "1"; "0" - means no people in the room; "1" - means somebody present in the room

SensorId	ChannelId	TimeStamp	Value
id1	ch1	2018-03-09T14:05:16	70.456
id1	ch2	2018-03-09T14:05:17	50
id2	ch1	2018-03-09T13:15:19	73.1
id2	ch2	2018-03-09T13:15:19	68
id2	ch3	2018-03-09T15:45:34	1
id3	ch1	2018-03-09T14:05:16	0

Output Data Source 1 (15 minutes time slots):

CSV file with following structure:

- TimeSlotStart – String; ISO Format of Local Date Time (without offset); The Start Date and Time of the 15 minute time slot. e.g:
  - 2018-03-09T14:00:00
  - 2018-03-09T14:15:00
  - 2018-03-09T14:30:00
  - 2018-03-09T14:45:00
  - 2018-03-09T15:00:00
  - ...

- LocationId – String; Id of a Room
- TempMin – minimum temperature in time slot; in Celsius; formatted string with 2 digits after separator (e.g "19.56"); empty if TempCnt = 0
- TempMax – maximum temperature in time slot; in Celsius; formatted string with 2 digits after separator (e.g "27.74"); empty if TempCnt = 0
- TempAvg – average temperature in time slot; in Celsius; formatted string with 2 digits after separator (e.g "25.74"); empty if TempCnt = 0
- TempCnt – count of temperature values
- Presence - "true" or "false"; "true" if there is at least one presence value = "1" in the time slot;
- PresenceCnt – count of positive presence readings (count of presence value = "1")

Example (not related to Input Data Sources above):

TimeSlotStart	Location	TempMin	TempMax	TempAvg	TempCnt	Presence	PresenceCnt
2018-03-09T14:15:00	room1	22.34	27.54	24.55	18	true	2
2018-03-09T14:30:00	room1	21.34	26.00	22.21	3	false	0
2018-03-09T15:00:00	room1				0	true	4
2018-03-09T14:00:00	room1	22.00	22.00	22.00	1	true	1
2018-03-09T14:00:00	room2				0	false	0
2018-03-09T14:45:00	room2	21.34	29.89	23.89	20	true	55

Time slots without data should be the next:

2018-03-09T14:45:00	room1				0	false	0

Output Data Source 2 (1 hour time slots):

CSV file with following structure:

- TimeSlotStart – String; ISO Format of Local Date Time(without offset); The Start Date and Time of the 1 hour time slot. e.g:
  - `2018-03-09T14:00:00`
  - `2018-03-09T15:00:00`
  - `2018-03-09T16:00:00`
  - `...`
- LocationId – String; Id of a Room
- TempMin – minimum temperature in time slot; in Celsius; formatted string with 2 digits after separator (e.g "19.56"); empty if TempCnt = 0
- TempMax – maximum temperature in time slot; in Celsius; formatted string with 2 digits after separator (e.g "27.74"); empty if TempCnt = 0
- TempAvg – average temperature in time slot; in Celsius; formatted string with 2 digits after separator (e.g "25.74"); empty if TempCnt = 0
- TempCnt – count of temperature values
- Presence - "true" or "false"; "true" if there is at least one 15 minutes time slot with presence = "true" in the 1 hour time slot;
- PresenceCnt – **in range from 0 to 4;** count of 15 minutes time slots with presence = "true";

Example (not related to Input Data Sources above):

TimeSlotStart	Location	TempMin	TempMax	TempAvg	TempCnt	Presence	PresenceCnt
2018-03-09T14:00:00	room1	22.34	27.54	24.55	45	true	2
2018-03-09T15:00:00	room1	21.34	26.00	22.21	34	false	0
2018-03-09T16:00:00	room1				0	true	4
2018-03-09T15:00:00	room2	22.00	22.00	22.00	23	true	1

Time slots without data should be the next:

2018-03-09T13:00:00	room2				0	false	0