Programming problems:

(1) Assume that you receive a feed in a "TrainDetails.txt" (you need to refer the flat file attached in the mail) from an external system and the file contains TrainType, Speed(KMPH), Energy(KWH). Note that each column/field is separated by delimiter '-' as given in the file. Can you write a stand-alone java program to read the file and load/store into in-memory (do not use any database) and can you display below details upon user enters the train type as the input?

(i) Print the lowest speed for the train type or print 'No details found' for invalid train type.
(ii) Print the highest energy consumption and the speed details or print 'No details found' for invalid train type.

(2) The below table contains the average passenger demand between Waterloo to Woking train stations during 08:00AM to 08:15AM:

Destination / **Origin**	Waterloo	Surbiton	Esher	Hersham	Walton-on-Thames	Woking
Waterloo	0	400	500	650	450	200
Surbiton	450	0	235	350	250	250
Esher	250	350	0	200	340	150
Hersham	150	230	220	0	310	350
Walton-on-Thames	230	150	320	225	0	320
Woking	250	340	95	245	230	0

Can you write a stand-alone java program to perform the below operations?

(i) Ask the user to input the above stations and their passenger demands (can be any values, above table is just an example) and load them to a matrix & print the matrix to a console.
(ii) Also, print (to console) the maximum demand and the station names. For example, for the above sample demand data entered by the user, the output should be like below: Maximum demand is between Waterloo to Hersham (650).
Note: If multiple records found, then print all of them.
(iii) At the end, ask the user to enter the "minimum demand for investigation" so that the program prints all the stations above that minimum demand.
For example, for the above sample demand data, if the user enters 450 as the "minimum demand for investigation" then the output should be like below:
Waterloo to Esher – 500
Waterloo to Hersham – 650
Waterloo to Walton-on-Thames – 450
Surbiton to Waterloo – 450
Note: If no stations found with that minimum demand, then print "No Results"

(3) Assume that you receive a feed in a "DriverAndDelayDetails.txt" (you need to refer the flat file attached in the mail) format from an external system for every 30 minutes time period. The feed provides the journey details of various trains running (in the past 30 minutes) across the UK rail network and the status for each train (typically the status could be either in progress or completed). The file contains train id (unique for each journey), station, driver name, departure lateness and each column is delimited by '|'. When the train reaches the destination stop, the departure lateness column will be indicated with 'NA' (not applicable), if you find a value other than 'NA', then the train journey is still in progress. Write a program to read the data

from the file and then load it into two different tables i.e., train_driver_details and train_delay_details. You are free to use any of your favorite databases (for example, MySQL, PostgreSQL, MongoDB, etc..) and any API of your choice like an ORM framework or Spring or plain JDBC etc. to connect to the database. Also, note that, if you are using SQL database, you need to send us the DDL script to create the tables.

The sample data loaded into the tables look like below (sample data):

train_driver_details table/collection:

Train Id	From Station	To Station	Driver Name	Journey Status
1A99	Newcastle	Carlisle	Matt Hudson	INPROGRESS
1C76	Carlisle	Metro Centre	Stephen Taylor	COMPLETED
1A99	Carlisle	Waterloo	John Warner	COMPLETED

train_delay_details table/collection:

Train Id	Station	Departure Time at station	Departure lateness in seconds
1A61	Waterloo	2018-09-04T14:32:00	0
1B87	Newcastle	2018-09-04T15:05:00	30
1U61	Guildford	2018-09-05T10:20:00	15

Important Instructions:

(1) The data provided for the programs is just a sample of few records, but in reality, your program needs to handle many thousands of records.
(2) Write only one stand-alone main Class for one problem. You can place all in a project and send us the project or share the GitHub link.
(3) You can use maven for your project and specify the required dependencies inside the pom.xml
(4) Strictly follow the Object Oriented (OO) design principles and best practices.
(5) Ensure that your code compiles & runs without any errors and produces the desired outputs.