IB Computer Science

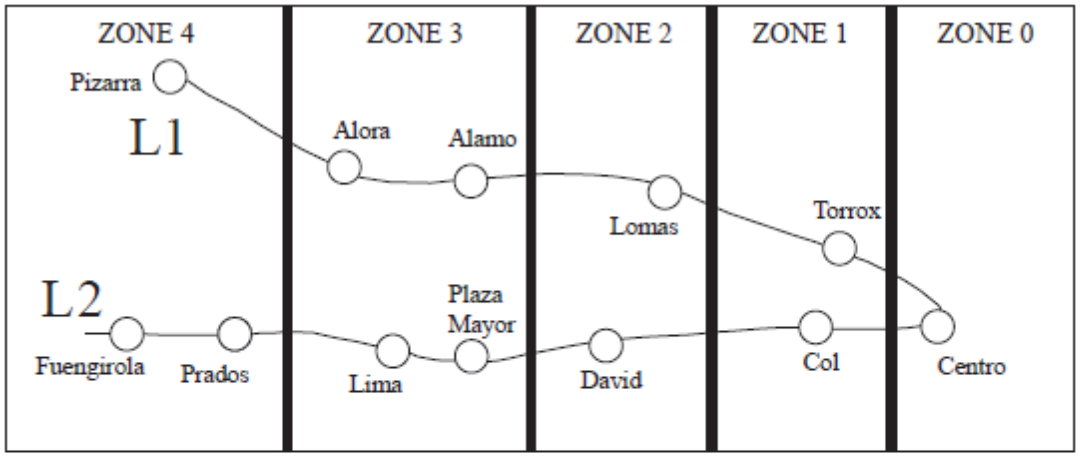
Student_____

Java Programming and Algorithms

Date: _____

Directions: Answer each part of each question. You have seen part of this problem before, now it is time to put all of these pieces together.

A suburban railway system for a large city in Southern Europe consists of two lines **L1** and **L2**, which meet at the station Centro, where passengers can change from one line to the other. The system is shown below.



Each station is located in a particular zone, and the total number of zones in which the journey takes place determines the train fare. Note, if a passenger starts in **Zone 1**, goes to **Zone 0** and then back to **Zone 1**, the journey has taken place in **three** zones. Examples of the number of zones are shown below for different journeys.

Travelling from	Travelling to	Number of zones
Lima	Plaza Mayor	1
Alora	Plaza Mayor	7
Lomas	Col	4

(a) State the number of zones in which the journey takes place when travelling from Alora to Fuengirola. *[1]*

Student_____

Java Programming and Algorithms

Date: _____

The data for each station (station name, line, zone) is stored on the system's server in the collection `TRAIN_DATA`. There are 12 stations in total. The first part of the collection is shown below.

```
Centro,  L1,  0,  Alora,  L1,  3,  Torrox,  L1,  1,  Col,  L2,  1,  ...
```

From this we can see that Alora is part of line L1 and is located in Zone 3.

At the start of each day, the data in `TRAIN_DATA` is read in to the binary tree `TREE`, in which each node will hold the data for one station. The binary tree will be used to search for a specific station's name.

(b1) In your Cloud9 _java workspace, create a data file named Train.data that contains one line of input for each station, using the format Station Name, Line, Zone. Put the data in the file in the following Station Order:

Centro, Alora, Torrox, Col, Lima, Lomas, David, Prados, Fuengirola, Alamo, Pizarra, Plaza Mayor

(b2) Sketch a binary tree created by reading the file in the order specified above.

IB Computer Science

Student_____

Java Programming and Algorithms

Date: _____

The TRAIN_DATA collection is also used to construct the one-dimensional array STATIONS (which only contains the list of station names sorted into alphabetical order), where STATIONS[0] = Alamo.

(c1) Write a java program that reads in the Train.data file and creates a collection named TRAIN_DATA.

(c2) Use the TRAIN_DATA collection created from (c1) above to construct a one-dimensional array named STATIONS, which contains a list of the station names sorted into alphabetical order. For example. STATIONS[0] is Alamo.

(c3) Create a method named PrintStations that print the STATIONS array. What is out output value for STATIONS[4] ?_____

The two data structures TRAIN_DATA and STATIONS are now used to construct a two-dimensional array named FARES containing the fares between stations (partly shown below). Note the fare for

travelling in **each** zone is €1.00.

FARES	Alamo	Alora	Centro	Col	...
Alamo	0	1.00	4.00	5.00	...
Alora	1.00	0	4.00	5.00	...
Centro	4.00	4.00	0	2.00	...
Col	5.00	5.00	2.00	0	...
...	...	...	...	...	... *etc*

(d1) Calculate (on paper) the fare for traveling from Torrox to Lima. _____

(d2) Calculate (on paper) the fare for traveling from Alora to David._____

Java Programming and Algorithms

(e) [on paper]

Construct the algorithm that would calculate the fares for this two-dimensional array. You can make use of the following two sub-procedures:

```
TRAIN_DATA.getZone(STATION)  // which returns the zone in which the
                             // station is located
TRAIN_DATA.getLine(STATION)  // which returns the line on which the
                             // station is located
```

Your algorithm should make as few calculations as possible. *[9]*

Student_____

Java Programming and Algorithms

Date: _____

(f1) Implement the method TRAIN_DATA.getZone(STATION)

```
// which returns the zone in which the
// station is located
```

(f2) Implement the methods TRAIN_DATA.getLine(STATION)

```
// which returns the line on which the
// station is located
```

(f3) Create a method of initializing the FARES table from the TRAIN_DATA and STATIONS data structure. (implement the algorithm you developed in part e) .

(f4) Create a method named Print_Fares that will print a completed version of the Fares table .

Hint: What is the first thing you should do?

(f5) Print a screen shot the output of the Print_Fares table.