**Homework 6: Hotel Reviews**

1 Objective

In this project, you’ll read in and manage hotel reviews from several reviewers. This will require you to create a two-dimensional array (**reviewers** x **hotels**) as well as parallel arrays to hold associated data. You’ll also implement a sorting algorithm which will require you not only to sort in the traditional way, but to maintain parallelism among data. There won’t be any exciting UI here, just some output of array data, and hotel/review data coming from a file.

2 Input File

2.1 File

The input file will contain the following elements, in order:

* On the first line, an integer indicating how many hotels were reviewed (***h***), followed by an integer indicating how many reviewers have provided data (***r***).
* On the next ***r***lines, a sequence of ***h*** integers, each a review score in the range 1 to 10.
* Once the review data is finished, the following ***h***lines will contain the hotels titles.
* Data elements are tokenized (separated by space (or spaces), tabs, etc.

2.2 Example Contents

**5 4**

**7 7 8 6 9**

**5 9 9 7 10**

**7 10 7 9 8**

**10 8 9 10 9**

**Grand Solmar Lands End Resort and Spa**

**Sandos Finisterra Los Cabos**

**Grand Fiesta Americana Los Cabos**

**Marina Fiesta Resort & Spa**

**RIU Santa Fe**

The example file’s contents are interpreted as follows:

* Four reviewers participated, reviewing five hotels.
* The first reviewer gave the first hotel “7”, the second hotel “7”, and the third “8” and so on. Next reviewers scores follow in similar fashion.
* The five hotels reviewed are titled “Grand Solmar Lands End Resort and Spa,” “Sandos Finisterra Los Cabos,” “Grand Fiesta Americana Los Cabos,” “Marina Fiesta Resort & Spa,” and “RIU Santa Fe” respectively.

3 Concepts

3.1 Read the Data

You’ll need to read the file’s hotel review contents into an array called ***reviews***. The hotel names will be in a parallel array called ***hotels***, i.e., column 0 in ***reviews*** matches item 0 in the ***hotels*** array.

3.2 Create an Average Ranking Array

Now you’ll create an array to hold average rankings per hotel (***avgRanks***) and populate it. In this example, the first hotel would average (7+5+7+10+)/4 = 7.25, the second 8.5, the third 8.25, the fourth 8.0, and fifth 9.0. This means you now have three parallel arrays: ***reviews***, ***hotels***, and ***avgRanks***.

3.3 Sort the Arrays by Average Ranking

You should sort tree arrays (***reviews***, ***hotels***, and ***avgRanks***). The hotel with the highest average ranking comes first and the one with the lowest ranking last. But remember to maintain parallel data as well; ***hotels*** and ***reviews*** need to move in lockstep with the ranked review data.

After sorting, you must get next result:

Array ***avgRanks***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 9 | 8.5 | 8.25 | 8 | 7.25 |

Array ***reviews***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 9 | 7 | 8 | 6 | 7 |
| 10 | 9 | 9 | 7 | 5 |
| 8 | 10 | 7 | 9 | 7 |
| 9 | 8 | 9 | 10 | 10 |

Array ***hotels***

|  |
| --- |
| RIU Santa Fe |
| Sandos Finisterra Los Cabos |
| Grand Fiesta Americana Los Cabos |
| Marina Fiesta Resort & Spa |
| Grand Solmar Lands End Resort and Spa |

4 Class

Create a HotelReviews class and use it for this project. No other class-level data should be needed, other than the private data shown.

|  |
| --- |
| **HotelReviews** |
| -hotels : Array of String |
| -reviews : 2D Array of Integer |
| -avgRanks : Array of Double |
| **Constructors** |
| +HotelReviews(File) |
| **Accessors** |
| +getHotelCount() : Integer |
| +getRankHotel(review : Integer, hotel : Integer) : Integer |
| +getHotel(index : Integer) : String |
| +getAvgRank(index : Integer) : Double |
| **Private Methods** |
| -readData(File) |
| -calculateAvgRankings() |
| **Other Methods** |
| +displayReviews() |
| +displayAvgRanks() |
| +displayHotels() |
| +sortByRanking() |
| +test() |

The **constructor should only** call methods readData and calculateAvgRankings (private methods).

5 Code Implementation

Create a **main** class which creates a new instance of the **HotelReviews** class, reads data from the file, and displays all arrays before and after sorting. See examples of output.

6 Testing

* Write test methods for each of your methods. You don’t need to test the display\* methods.
* Test with different numbers of hotels and reviewers to make sure your algorithms are correct.
* Manual inspection will be necessary; the debugger is your friend and will be very useful here.

7 Submitting Your Work

* Submit your .jar or .zip file
* If your test method relies on test files, be sure and include them in your submission.

Examples of Output

**Row Reviews**

7 7 8 6 9

5 9 9 7 10

7 10 7 9 8

10 8 9 10 9

**Average Rankings**

7.25 8.5 8.25 8.0 9.0

**Hotels**

Grand Solmar Lands End Resort and Spa

Sandos Finisterra Los Cabos

Grand Fiesta Americana Los Cabos

Marina Fiesta Resort & Spa

RIU Santa Fe

**Ranked Reviews**

9 7 8 6 7

10 9 9 7 5

8 10 7 9 7

9 8 9 10 10

**Hotels Rating**

RIU Santa Fe 9.0

Sandos Finisterra Los Cabos 8.5

Grand Fiesta Americana Los Cabos 8.25

Marina Fiesta Resort & Spa 8.0

Grand Solmar Lands End Resort and Spa 7.25