Arrays and files in Visual Basic II

Topics

- Arrays and parallel arrays
- Reading from CSV files into parallel arrays
- More practice with formatting strings
- Finding the minimum value in an array
- Using the `Arrays.sort` function in Visual Basic

Preliminaries

Go to today's lab folder (`Lab06_Arrays_files`) on the `N:` drive and copy the write-up for today (**not the whole folder**), `Lab-II.pdf`, to your `M:\CS130\Labs\Lab05_YourLastName_YourFirstName` folder. You may now work locally by opening the write-up from within the copied folder.

Rating predictions

5. Use **File Explorer** to navigate to your lab folder. Copy and paste the folder `RatingPredictions_Part4`, and then right-click on the copy to rename it as `RatingPredictions_Part5`. Launch VS Express and use it to open the project `RatingPredictions_Part5`.

   TODO Better description with reference to book on how to find minimum. Using a `For`-loop, find the minimum value in the `distance` array. Be sure to keep track of the index where the minimum value appeared. Then, using the minimum distance and its corresponding index, make a prediction as to the rating for the user for movie five. For now, the prediction will be the rating for movie five of the user from the file that is the least different (i.e., minimum distance) from the user's ratings for movies one through four. Output the predicted rating in a message based on the following example:

   ```
   The most similar user was user #XXX and the distance calulcated was X.X.
   The predicated rating for movie five is X.X.
   ```

   After you get your program working, fill out `Test Table 3` in the file `RatingPredictions_TestTables.docx`, making sure that your expected output and actual output are consistent.

Your project should behave identically to the solution which can be run by double-clicking the file `RatingPredictions_Part5.exe` found in the `Executables` folder inside of your lab folder.

When done, save and close project `RatingPredictions_Part5`.

Checkpoint 5 (80/100): A successful `RatingPredictions_Part5` project:

- produces correct output for all test cases in Test Table 3
- outputs the prediction in a statement following the example given above

6. Use **File Explorer** to navigate to your lab folder. Copy and paste the folder `RatingPredictions_Part5`, and then right-click on the copy to rename it as `RatingPredictions_Part6`. Launch VS Express and use it to open the project `RatingPredictions_Part6`.

   As you may be able to guess, the simple model of finding the user from the dataset that is the least different from the user's input ratings and using that user's rating for movie five to predict the user's input for movie five is not very robust, i.e., it is likely to produce incorrect predictions. However, your work up to this point is not for naught. Rather than using just the dataset sample that is least different from the user's sample, we can greatly improve the accuracy of our predictions by looking at the k least different dataset users.

   Unfortunately, finding the k least different dataset users is more complicated than find just one, so as before, we will work through it in steps. The basic idea is to sort the `distance` array and then select the first k values from the array as the will represent the k minimum distances.

   To see how this works, we must first sort the `distance` array. To do this, we will rely on a built-in function in Visual Basic, conveniently named `Array.Sort`. So, to sort the `distance` array, you would say:

```
Array.Sort(distance)
```

   Output the sorted `distance` array to `outResults` to see that it is in fact sorted.

   TODO Have students work on student name problem to see how the two parameter variant of `Array.Sort` can help them.

   Unfortunately, using the above line of code, after sorting the `distance` array we have no way to determine which dataset sample each of the distance values belonged to. Fortunately, with a small modification, we can have `Array.Sort` produce a second array which stores the sample number for each value in the sorted `distance` array. To do this, we must supply a second array to the `Array.Sort` function. We will call this array `userId` and it will contain the user id that is associated with each distance value.

```
Array.Sort(distance, userID)
```

   Now we can find out which user belongs to each distance in the `distance` array like so:

```
outResults.AppendText("The distance between user #" & userIDs(0) & _
    " and the user's sample is " & distance(0))
```

   Now, reproduce the prediction from the previous part, i.e., predict based on the rating of the dataset user that is the least different from the user's ratings. This time however, use the sorted array instead of computing the minimum distance in a loop. Output your prediction using the same statements as above.

   Test your program by checking that it is producing correct results for all of the test cases in Test Table 3.

Your project should behave identically to the solution which can be run by double-clicking the file `RatingPredictions_Part6.exe` found in the `Executables` folder inside of your lab folder.

When done, save and close project `RatingPredictions_Part6`.

Checkpoint 6 (90/100): A successful `RatingPredictions_Part6` project:

- produces correct output for all test cases in Test Table 3, using a sorted distance array
- outputs the prediction in a statement following the example given above

7. Use **File Explorer** to navigate to your lab folder. Copy and paste the folder `RatingPredictions_Part6`, and then right-click on the copy to rename it as `RatingPredictions_Part7`. Launch VS Express and use it to open the project `RatingPredictions_Part7`.

   Ask the user to input an integer, k, that will be used to choose the k closest matches to the user's sample. Using the sorted `dataset` array and the `sampleNumber` array, output the k least different dataset samples in a table formatted as follows:

8. Use **File Explorer** to navigate to your lab folder. Copy and paste the folder `RatingPredictions_Part6`, and then right-click on the copy to rename it as `RatingPredictions_Part7`. Launch VS Express and use it to open the project `RatingPredictions_Part7`.

   Find the average of the top k values and use this as a prediction.